参赛密码 ____________

（由组委会填写）

"华为杯"第十四届中国研究生

数学建模竞赛

学　　校	清华大学
参赛队号	K0263
队员姓名	1. 赵笑阳
	2. 龚欢
	3. 张兴艺

参赛密码 ____________

（由组委会填写）

“华为杯”第十四届中国研究生
数学建模竞赛

题 目 **构建地下物流系统网络**

摘 要：

随着城市人口的增加和城市规模的扩张，交通问题越来越频繁地困扰着人们。城市中每天大量的货物运输在交通拥堵问题中难辞其咎。因此，城市地下物流系统的建立势在必行。本文以南京市仙林区为例，以解决交通拥堵为目标，对现代城市地下物流系统网络的构建问题进行了由浅及深、循序渐进的研究。

在问题一中，本文首先对地下物流网络的节点选择进行了研究。这一研究主要分两步进行：首先，我们将节点选择问题转化为混合整数规划模型，借助 matlab 和 lingo 求出最优解，得到能够达成解决交通拥堵目标的最少节点的数目及其分布。在该问题中，最少的节点数目为 27，分别位于区域 791、794、795、806、807、808、815、818、824、825、841、844、851、858、865、867、871、872、873、875、879、881、888、889、894、899、900 的中心位置。而后，我们将一级节点的选择问题转换为带约束的最优化问题，以降低园区一级节点转化率为目标从节点中选择出最合适的四个作为一级节点，分别是 791、807、867、889 号区域中心。选择出一级节点后，本文根据就近原则将二级节点分配给四个一级节点作为子节点。

在问题二中，本文基于问题一求得的一、二级节点配置，以日均总成本最低为目标对地下物流系统的线路和通道类型进行了规划。总成本分成固定折旧成本

和流量成本。我们按照第一问中确定的一级节点及其附属二级节点将整个仙林地区分为四个大型区域，在区域内通过最小生成树法求得连接各个节点的最短路径，在区域间综合考虑道路长度和流量来规划一级节点间的线路，并根据各条线路的实际流量来选择最合适的隧道类型，确定了由 27 个节点和 30 条通道构成的地下物流系统网，并求得方案的总成本为 248.13 万元。

在问题三中，我们对前文中分步规划的地下物流系统节点和线路进行了统筹的优化改进。首先，我们从节约成本的角度对该物流系统网络进行优化，以总成本最低为原则重新从 27 个节点中选取一级节点，得到了日均成本 228.6 万元的地下物流系统设计方案。而后，我们分别针对面对货运量激增和面对线路中断的情况提升了系统抗风险性能，具体为通过提高设计系统的容纳能力阈值和增加环线建设的角度来实行。

在问题四中，我们考虑了分 8 年建设供未来 30 年使用的地下物流系统建设方案。由于时间有限，本文没有对 8 年建设的具体方案进行详述，而是提出了解决该问题的思路和原则，并进行了一定预估。首先，本文对未来 30 年的交通需求、状况和设施能力进行了初步估计；而后讨论了解决前三个问题的方法在当前问题情况中的适用与否，提出了增加一级节点的基本判断，并指出可以借助核密度估计方法和采用 fleury 算法寻找欧拉回路的方法来确定节点分布和线路规划；最后，本文引入节点重要度的概念并从动态的角度为将工程量分八年完成，提出了评判重要度的三条准则。

关键词：地下物流系统，集合覆盖，混合整数规划，最小生成树，分步规划

目录

一、问题背景与重述

1.1 问题背景和提出

衣、食、住、行自古以来就是人们最为基本的需求，随着人类社会的高速发展，城市规模不断扩大，人们分工不断细化，人对出行和货物运输的需求都在不断地增加，因此现代城市中的交通也面临着越来越大的压力。现有的城市道路运载能力常常已经难以满足日益增长的交通需求，交通拥堵每一天都困扰着各个大城市的政府和居民。

交通拥堵会对城市生活造成多方面的影响：首先，交通拥堵会对人的时间造成大量浪费，从而对经济活动造成很大影响；其次，交通拥堵会在很大程度上增加车辆燃油的消耗，对生态环保造成不利影响；此外，交通拥堵会影响人的情绪，增大事故发生的可能性，对人们的心理健康也有一定影响。

交通拥堵主要是由于道路容量不足、汽车使用频率增加、道路交汇处过多等因素造成的。地下物流系统的使用则可以对这三方面进行有效改进。地下物流系统(ULS)指城市内部及城市间通过类似地铁的地下管道或隧道运输货物的运输和供应系统。通过使用地下物流系统进行货物运输，可以减少路面汽车的使用，缓解道路交通的压力；同时，将交通由二维升级为三维，可通过合理规划来大量减少交汇带来的拥堵问题。此外，由于地下物流系统通常使用自动导向车，可以节省大量劳动力。因此，地下物流系统的发展是当前时代发展的必要需求。

虽然地下物流系统的使用势在必行，但目前为止，对地下物流系统的研究才刚刚起步，只有一些小范围内的试点运营，缺乏系统而完善的理论模型指导。本文以南京市仙林区为例，试图构建适用于该地区的地下隧道货物运输的物流系统网络，以尽可能低的成本通过地下物流分流使地面交通保持通畅。

1.2 问题重述

问题一：地下物流节点的选择。

在尽可能节约资源的情况下（节点数目尽可能少，一级节点转运率 φ 尽可能低），给出能够有效疏通（使得交通拥堵指数不高于 4）各个物流区域交通拥堵的地下物流一、二级节点配置，并计算出各个节点的服务范围、实际货运量和各一级节点的转运率。

问题二：地下通道网络设计。

在尽可能节约资源的情况下（货物运输成本和物流隧道、节点建设成本和折旧损失之和尽可能小），基于上题布置的物流网络节点群，适度调整各级节点位置，并设计合适的地下路线，得出各个节点新的位置、实际货运量和各级通道的位置和实际流量。

问题三：网络改进

从全局对分两步设计的地下物流网络系统进行改进，以期缩短运输里程、降低运输成本、增加地下物流系统的抗风险能力等。

问题四：建设时序与动态优化

设计分八年完成的“地下物流网络系统”建设方案以满足南京市仙林地区近 30 年以每年 5%速度上涨的交通需求，并将该方案形成的物流网络与问题三中的物流网络进行对比。

二、模型假设与符号说明

2.1 模型假设

为了对问题进行简化以方便计算和研究，本文中进行以下假设：

1）假设拥堵指数仅根据货运量计算；

2）若节点覆盖了某区域中心点即假设为对该区域进行了覆盖；

3）假设货物返回地面后采用人力或小型车辆在节点服务区域内进行运输不影响交通；

4）如果一个交通货运区域上存在物流节点，则节点建造于该区域的中心位置；

5）近似认为区域交通拥堵指数与 OD 数据反映出来的区域总货运量（进、出之和）成正比；

6）忽略车辆的长度，将车辆看为质点；

2.2 概念和符号说明

相关概念：

- 实际吞吐量：一个交通货运区域实际的进、出货运量总和称为该区域的实际吞吐量。
- 饱和吞吐量：使得一个区域地面交通状况仍处于“基本畅通”的最大吞吐量称为该区域的饱和吞吐量。
- 额外吞吐量：一个交通货运区域当前的实际货运量超过饱和吞吐量的部分，即需要通过地下物流系统分担的货运量，称为额外吞吐量。
- 园区地上出货量：园区发出的货物中直接通过地上运输出货的部分。
- 园区一级节点非转运量：园区通过地下物流系统发往最近的一级节点，或通过该一级节点发往其附属的二级节点的货运量。
- 园区一级节点转运量：园区通过地下物流系统经由最近的一级节点发往其他一级节点及其附属的二级节点的货运量。
- 大型区域：一个一级节点及其下辖的二级节点服务的总范围为一个大型区域，大型区域不包含园区，与园区一一对应。

符号说明：

概念	符号	概念	符号
物流节点	a_j	单程列车运行时间	t
交通货运区域	b_i	交通拥堵指数	k_i
区域的现吞吐量	S_i	物流节点集合	M
交通货运区域集合	N	距离	$L(i,j)$
候选节点的决策函数	X_j	节点服务能力	D_j
节点对区域额外吞吐量的分担比例	Y_{ij}	园区至最近一级节点的通道长度	T

三、问题分析

3.1 针对问题一

问题一的任务为在尽可能节约资源的情况下，给出能够有效疏通各个物流区域交通拥堵的地下物流一、二级节点配置，并计算出各个节点的服务范围、实际货运量和各一级节点的转运率。由于问题一中暂时不考虑货物运输成本和线路建设、折损成本等，因此节约资源即表现为节点数目尽可能少、一级节点转运率φ尽可能低。而有效疏通各个物流区域交通拥堵的标志为使得仙林区所有交通货运区域的交通状况达到“畅通”或“基本畅通”，即其交通拥堵指数不高于 4。因此，我们可以考虑将问题一转换为带约束的最优化问题。

我们首先计算出各个交通货运区域的额外吞吐量（额外吞吐量定义见前），对于任何节点，假定其服务半径为 3km，与地上交互量为 3000 吨，我们的目的是最小化节点数目，故将问题转换为带约束的混合整数规划问题，并通过 lingo 和 matlab 分别求出最优解，即二级节点的选取。选定二级节点后，我们优化了一级节点的选取，达到令各园区转运率最小化的目的。

3.2 针对问题二

问题二的任务为在尽可能节约资源的情况下，基于上题布置的物流网络节点群设计合适的地下路线，得出实际货运量、各级通道的位置和实际流量。

我们的目的是最小化成本，而成本包含两个部分：流量成本（货物运输成本）和固定折旧成本（节点和隧道建设的日均成本）。流量成本取决于货物量及每单位重量的货物运输的距离，可以分为从园区到该园区所对应一级节点的货物运输、一级节点及其下辖二级节点之间的货物运输和一级节点之间的货物运输三部分来计算；固定折旧成本包括节点建设成本、园区到该园区所对应一级节点隧道成本、一级节点及其下辖二级节点间隧道成本和一级节点之间的隧道成本。由于节点和通道的设计年限均为 100 年，日均成本为其总建设成本的$\frac{1}{365\times100}$。在第一问已经确定各级节点和一二级节点隶属关系的情况下，为了让固定折旧成本最低，我们对各一级节点及其下辖二级节点所构成的图采用最小生成树法，求出满足流量约束的最短连通路径，并检验各隧道流量以判断采用双轨还是四轨隧道。之后我们综合考虑道路长度和流量来确定一级节点间的路线，并按照实际流量为各路线选择合适的通道。最后将各部分成本累加得到每日的总成本。

3.3 针对问题三

问题三的任务为从全局考虑来对问题一、二中建立的地下物流网络进行优化改进。改进体现在两个方面：1）进一步降低资金成本和时间成本；2）增加地下物流网络的抗风险能力。

对于进一步降低成本这一任务而言，该问题可以首先从第一、二问在设计目标、条件限制等方面的差异出发，寻找从全局来看可以优化之处。具体而言，我们仍采用分步规划的方式，首先仍通过解决混合整数规划问题的方法确定 27 个节点，但与问题二不同的是，问题二已经在步骤一中根据最小化转运率的标准确定了一级节点，而我们把一级节点的选择放在步骤二，相当于在考虑最小费用的时候，加入了一级节点的选择，从这个角度而言，比起问题二的方法，我们从更

全局的角度考虑问题，因而也自然而然的得到了比问题二最优解更加低的总成本。

在从全局着眼对地下物流网络系统进行降低成本优化后，再考虑增加地下物流系统的抗风险能力。地下物流系统的抗风险能力主要表现为面对某方向货运量激增时能够有较大的容纳能力和其面对某通道中断时受影响范围较小。对于前者，我们重新规划了节点个数、位置、划分和隧道布局，给隧道流量、节点与地面进出额度均留下了一定安全空间，使得整体系统可以面对货运量增加的情况。对于后者，我们在前者的基础上，适当增加了隧道数目，在进一步分散流量的同时，减少了某隧道突然中断的影响。

3.4 针对问题四

问题四假设未来 30 年需求量每年呈 5%增长，我们不妨假设通过新增道路，拓宽现有道路等措施，地面道路的交通运输能力也以每年 5%的增长率增长。因此各个物流区域需要疏通的交通额度都变为原来的 1.05^30=4.32 倍。在此设定下，若我们继续假设各节点从地面收发货物总量上限为 3000 吨，则原问题无可行解，这是因为即使把所有 110 个中心区域都设为节点，其与地面的交互能力也小于所需要缓解的交通额度。故此，我们考虑到节点修建技术也在增长，假定各二级节点从地面收发货物总量上限增长到 6000 吨，一级节点增长到 8000 吨，同时增加了一个一级节点。在此假设下，采用分步规划方法求出最优解（最小化费用）。考虑到建设时序，我们优先建设园区到各自一级节点间的隧道，然后建设一级节点间的隧道；对于一级节点及其下辖二级节点内部区域，优先修建重要度高的节点（以经过此节点流量多少及其是否和已有隧道相连作为评判标准）。第三题设计的抗风险网络将于四年后失效，因此建议采用第四题的节点及隧道的布局和修建方法。

四、问题一的解答

4.1 各区域的额外吞吐量

我们将一个交通货运区域实际的进、出货运量总和称为该区域的**实际吞吐量**。在一个区域面积、地面设施等情况不变的情况下，实际吞吐量越高，该区域的交通状况越拥堵。

我们将使得这个区域地面交通状况仍处于“基本畅通”的最大吞吐量称为该区域的**饱和吞吐量**。当实际吞吐量低于饱和吞吐量时，该区域交通状况为“畅通”或“基本畅通”，当实际吞吐量高于饱和吞吐量时，该区域的交通状况为“轻度拥堵”“中度拥堵”或“高度拥堵”。

我们用 $b_1,b_2,b_3,\ldots b_i,\ldots b_{110}(i\leq 110)$分别来表示这些交通货运区域。现已知各个交通货运区域之间的进、出货运量和交通拥堵指数。我们将区域 b_i 当前的进、出货运量总和表示为 S_i，将区域 b_i 当前的交通拥堵指数表示为 k_i,由于可以近似地认为区域交通拥堵指数与 OD 数据反映出来的区域总货运量（进、出之和）成正比，所以可以求得区域 b_i 的饱和吞吐量为$\frac{4S_i}{k_i}$.

我们将一个交通货运区域当前的实际货运量超过饱和吞吐量的部分，即需要通过地下物流系统分担的货运量，称为额外吞吐量。区域 b_i 的额外吞吐量为：

$$额外吞吐量=\begin{cases} S_i-\dfrac{4S_i}{k_i}\ (当实际吞吐量>饱和吞吐量); \\ 0\ (当实际吞吐量\le 饱和吞吐量). \end{cases}$$

通过计算可以得到，仙林区各个交通货运区域的额外吞吐量如下表所示：

表 4.1.1 各交通货运区域的额外吞吐量

791	792	793	794	795
351.8349643	588.664172	714.2704065	561.5457044	327.0350225
796	797	798	799	800
596.5036457	727.2309398	659.4549528	589.6048081	359.2558498
801	802	803	804	805
640.4432987	571.0252529	447.1331378	198.2530789	488.7035452
806	807	808	809	810
267.7667711	803.5175512	1005.897079	315.975	605.7361852
811	812	813	814	815
383.0766598	1060.87219	935.5764098	634.5416991	1452.998007
816	817	818	819	820
175.0110573	363.4281398	459.4842218	998.6510562	1293.83986
821	822	823	824	825
1226.753319	1294.372076	1016.523745	439.7193245	983.8155463
826	827	828	829	830
1318.603937	993.5555662	222.9463744	501.487918	1137.476705
831	832	833	834	835
674.5886383	432.4395508	785.8310725	843.0324286	1258.622949
836	837	838	839	840
308.3556248	851.9075537	655.3763295	2267.640495	746.6497974
841	842	843	844	845
528.4242198	536.5345714	734.4849426	686.4635796	646.906116
846	847	848	849	850
821.7511901	725.2454482	1117.068689	425.4225652	484.7578095
851	852	853	854	855
471.3221765	705.9031515	549.4114554	466.9363582	533.4045039
856	857	858	859	860
384.6495351	367.8939343	473.445557	842.8996835	492.002064
861	862	863	864	865
605.383022	472.2936522	996.4749521	217.4354622	427.432511
866	867	868	869	870
0	1292.65258	682.7493844	540.8621139	343.7651464
871	872	873	874	875
259.6048485	225.806131	219.3092247	82.6233399	793.0045348
876	877	878	879	880

573.6921313	1085.422895	330.6964865	112.5025946	2397.451362
881	882	883	884	885
264.5177833	431.7718345	326.6300513	629.619754	331.6724671
886	887	888	889	890
2291.827698	531.9651293	697.8439879	604.2784598	1342.454762
891	892	893	894	895
424.1167895	1635.499809	1562.743407	1092.07401	0
896	897	898	899	900
0	371.9840488	1390.813761	1284.322868	789.3223131

只要地下物流系统承担了这一部分货运量，我们即可认为，地下物流系统疏通交通拥堵的目标达成了。

4.2 节点的约束条件——集合覆盖模型

与文章[11]相同，我们将此物流选址问题简化成集合覆盖问题。如上面所述，我们用 $b_1,b_2,b_3,\ldots b_i,\ldots b_{110}(i\leq110)$分别来表示编号自 791 至 900 的这 110 个交通货运区域，并将它们的集合定义为 N.

假设有 m 个可能被选定作为一、二级物流节点的备选位置，我们分别用 $a_1,a_2,a_3,\ldots a_j,\ldots a_m.(j\leq m)$来表示它们，并将这些位置的集合定义为 M. M 集合中的任意一个位置 a_j 处可能设置物流节点，也可能不设置物流节点。由于本题假设若节点覆盖了某区域中心点即假设为对该区域进行了覆盖，我们可以假设在某区域内的节点都位于该区域中心，即 M=N；

定义一个函数 X_j 来表示位置 a_j 处是否设置物流节点：如果在 a_j 处设置物流节点，则 X_j 的取值为 1；若在 a_j 处不设置物流节点，则 X_j 取值为 0：

$$X_j=\begin{cases}1\ (\text{当物流节点在}a_j\text{处时});\\0\ (\text{当物流节点不在}a_j\text{处时}).\end{cases}$$

定义集合 A(j)表示所有被物流节点 a_j 覆盖的交通货运区域中心构成的集合，另定义集合 B(i)表示所有覆盖交通货运区域中心 b_i 的物流节点构成的集合。

设 Y_{ij} 为区域中心 b_i 的额外吞吐量中由节点 aj 分担的比例。设 d_i 为区域中心 b_i 被地下物流系统分担的额外吞吐量（即表 4.1.1 中的值），设 D_j 为物流节点 a_j 可承担的吞吐量总量。

由题意可知，对于上述概念的取值存在一些约束：

1）如果一个物流节点 a_j 与一个交通货运区域存在服务关系，即物流节点的服务范围覆盖交通货运区域的中心点 b_i，则它们之间的距离（定义为 L(i.j)）应当不超过 3 千米，即对$\forall b_i\in A(j)$或对$\forall a_j\in B(i)$，有 $L(i,j)\leq3$；

2）覆盖一个交通货运区域中心的各个物流节点分担该区域中心的货运量总和刚好等于该区域中心被地下物流系统分担的吞吐量，即当$b_i\in N$且$a_j\in B(i)$时，各 Y_{ij} 的总和为 1：$\sum_{j,a_j\in B(i)}Y_{ij}=1\ \ (b_i\in N)$;

3）任意一个物流节点对其服务范围所覆盖的各个交通货运区域的额外吞吐量分担部分之和不超过该物流节点的总服务能力。当 a_j 处存在物流节点，其最大服务能力为 D_j（一级节点为 4000 吨，二级节点为 3000 吨）；当 a_j 处不存在物流

节点，其服务能力为 0 吨。由于

$$X_j=\begin{cases}1\ (\text{当物流节点在}a_j\text{处时})\\0\ (\text{当物流节点不在}a_j\text{处时})\end{cases}$$

可知，任何一个可能存在节点的位置 a_j 处的最大服务能力为 D_jX_j.

因此有：

$$\sum_{i,b_i\in A(j)} d_iY_{ij}\le \mathrm{D}_jX_j\ (a_j\in M)$$

4）由于 Y_{ij} 为区域中心 b_i 的额外吞吐量中由节点 a_j 分担的比例，因此有 $0\le Y_{ij}\le 1$，$1\le i\le 110$, $1\le j\le m$.

为方便表述，将上述约束条件 1）2）3）写成矩阵的形式 $K\cdot\mathbf{Z}\le\mathbf{b}$，其中 K 为矩阵，**Z**、**b** 为向量。

矩阵 K 为 2n 行 n(n+1)列的矩阵，其中第 1 至 n 行中：1 至 n 列均为 0；n+1 至 2n 列左上角到右下角的对角线上分别为 $A_{11},A_{21}\ldots A_{n1}$，其余为 0；2n+1 至 3n 列左上角到右下角的对角线上分别为 $A_{12},A_{22}\ldots A_{n2}$，其余为 0；以此类推， n^2+1 至 n(n+1)列自左上角到右下角的对角线上分别为 $A_{1n},A_{2n}\ldots A_{nn}$，其余为 0。其中矩阵 A 定义为

$$A_{ij}=\begin{cases}1\ (\text{当 L(i, j)}\le 3\text{km 时})\\0\ (\text{当 L(i, j)}>3\text{km 时})\end{cases}$$

K 的第 n+1 至 2n 行中：1 至 n 列左上角到右下角上均为-3000，其余处为 0；n+1 至 2n 列第 n+1 行从左到右分别为 $A_{11}d_1,A_{21}d_2\ldots A_{n1}d_n$，其余行均为 0；2n+1 至 3n 列第 n+2 行从左到右分别为 $A_{12}d_1,A_{22}d_2\ldots A_{n2}d_n$，其余行均为 0；以此类推，$n^2+1$ 至 n(n+1)列第 2n 行从左到右分别为 $A_{1n}d_1,A_{2n}d_2\ldots A_{nn}d_n$，其余为 0。

$$\begin{array}{cc|ccccccccccccccccc|}
\multicolumn{19}{c}{\text{n(n+1)列}}\\
\multicolumn{2}{c}{} & \multicolumn{4}{c}{\text{----n列----}} & \multicolumn{4}{c}{\text{----n列----}} & \multicolumn{4}{c}{\text{----n列----}} & \cdots & \multicolumn{4}{c}{\text{----n列----}}\\
 & & 0 & 0 & \cdots & 0 & A_{11} & 0 & \cdots & 0 & A_{12} & 0 & \cdots & 0 & & A_{1n} & 0 & \cdots & 0\\
 & \text{n行} & 0 & 0 & \cdots & 0 & 0 & A_{21} & \cdots & 0 & 0 & A_{22} & \cdots & 0 & \cdots & 0 & A_{2n} & \cdots & 0\\
 & & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \cdots & \vdots & & \vdots & \ddots & \cdots & \vdots\\
\text{2n行} & & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & A_{n1} & 0 & 0 & \cdots & A_{n2} & & 0 & 0 & \cdots & A_{nn}\\
 & & -3000 & 0 & \cdots & 0 & A_{11}d_1 & A_{21}d_2 & \cdots & A_{n1}d_n & 0 & 0 & \cdots & 0 & & 0 & 0 & \cdots & 0\\
 & \text{n行} & 0 & -3000 & \cdots & 0 & 0 & 0 & \cdots & 0 & A_{12}d_1 & A_{22}d_2 & \cdots & A_{n2}d_n & & & & \cdots & \\
 & & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \cdots & \vdots & \vdots & \ddots & \cdots & \vdots & \cdots & \vdots & \ddots & \cdots & \vdots\\
 & & 0 & 0 & \cdots & -3000 & 0 & 0 & \cdots & 0 & & & \cdots & & & A_{1n}d_1 & A_{2n}d_2 & \cdots & A_{nn}d_n
\end{array}$$

向量 $\mathbf{Z}=(X_1, X_2,\ldots X_n, Y_{11}, Y_{21},\ldots Y_{n1}, Y_{12}, Y_{22},\ldots Y_{n2},\ldots Y_{1n}, Y_{2n},\ldots Y_{nn})$为一个（$n+n^2$）*1 的向量，前 n 行分别为 $X_1, X_2,\ldots X_n$，第 n+1 至 2n 行分别为 $Y_{11}, Y_{21},\ldots Y_{n1}$，第 n+2 至 3n 行分别为 $Y_{12}, Y_{22},\ldots Y_{n2}$，第 n^2+1 至 n^2+n 行分别为 $Y_{1n},Y_{2n},\ldots Y_{nn}$。

向量 $\mathbf{b}=\begin{pmatrix}-1\\\vdots\\-1\\0\\\vdots\\0\end{pmatrix}$是一个 2n*1 的向量，前 n 行为-1，后 n 行为 0。

因此所有同时满足条件 $K\cdot\mathbf{Z}\le\mathbf{b}$ 和 $0\le Y_{ij}\le 1$，$1\le i\le 110$, $1\le j\le m$ 的解 Z 即是能够

将仙林区各个交通货运区域的交通拥堵状况疏导至“畅通”或“基本畅通”状况的可行解空间。

如上模型被称作集合覆盖模型，所谓集合覆盖模型（Set Covering model），就是对于需求已知的一些需求点， 设立相应的目标函数以及约束条件， 确定一组物流设施来满足这些需求点的需求量。具体的说，就是确定物流设施的最小数量和合适的位置。集合覆盖模型在商业物流系统的选址问题中应用较为广泛， 优点是它不仅能够用最少的物流设施点覆盖所有的用户需求，而且由于地下物流系统的建设难度以及投资费用远远高于地面建设，所以在一定程度上能够减轻地下物流系统建设初期巨大的投资费用和建设难度。

4.3 寻找最节约资源的节点分布方案——混合整数规划问题

在上一节中我们探讨了节点分布方案要达到疏解交通拥堵的目的需要满足的约束条件，这一节中我们将从中选择最节约资源的节点分布方案。

由于在问题一中我们暂且不考虑车辆和线路的成本，因此最节约资源的方案即节点数目最少的情况，此外尽可能保证园区的一级节点转运率低。

求解如上集合覆盖模型的最优解，即为在满足条件 $K\cdot\mathbf{Z}\leq\mathbf{b}$ 和 $0\leq Y_{ij}\leq 1$，$1\leq i\leq 110$，$1\leq j\leq m$ 的情况下求$min\sum_{j\in M}X_j$，这是一个混合整数规划问题。混合整数规划问题是一个 NP 难问题，解此问题的常用方法有分支定界法、割平面法或贪心算法等启发式算法，见文章[1]-[4]，我们采用 matlab 和 lingo 两种软件的自带算法来求解此问题，并且得到了互相印证的答案：最优解均为需要一、二级节点共计 27 个。

下面给出 matlab 计算的详细结果：

由于我们已经假设所有可能的节点都位于交通货运区域中心，因此可考虑各个区域中心 a_j 处是否建立物流节点。计算结果如下表所示：

表 4.3.1 matlab 的计算出的节点最优分布结果

a_j	791	792	793	794	795	796	797	798	799	800	801
j	1	2	3	4	5	6	7	8	9	10	11
X_j	1	0	0	1	1	0	0	0	0	0	0
是否建立物流节点	是	否	否	是	是	否	否	否	否	否	否
a_j	802	803	804	805	806	807	808	809	810	811	812
j	12	13	14	15	16	17	18	19	20	21	22
X_j	0	0	0	0	1	1	1	0	0	0	0
是否建立物流节点	否	否	否	否	是	是	是	否	否	否	否
a_j	813	814	815	816	817	818	819	820	821	822	823
j	23	24	25	26	27	28	29	30	31	32	33
X_j	0	0	1	0	0	1	0	0	0	0	0
是否建立物流节点	否	否	是	否	否	是	否	否	否	否	否
a_j	824	825	826	827	828	829	830	831	832	833	834
j	34	35	36	37	38	39	40	41	42	43	44
X_j	1	1	0	0	0	0	0	0	0	0	0
是否建立物流节点	是	是	否	否	否	否	否	否	否	否	否
a_j	835	836	837	838	839	840	841	842	843	844	845

j	45	46	47	48	49	50	51	52	53	54	55
X_j	0	0	0	0	0	0	1	0	0	1	0
是否建立物流节点	否	否	否	否	否	否	是	否	否	是	否
a_j	846	847	848	849	850	851	852	853	854	855	856
j	56	57	58	59	60	61	62	63	64	65	66
X_j	0	0	0	0	0	1	0	0	0	0	0
是否建立物流节点	否	否	否	否	否	是	否	否	否	否	否
a_j	857	858	859	860	861	862	863	864	865	866	867
j	67	68	69	70	71	72	73	74	75	76	77
X_j	0	1	0	0	0	0	0	0	1	0	1
是否建立物流节点	否	是	否	否	否	否	否	否	是	否	是
a_j	868	869	870	871	872	873	874	875	876	877	878
j	78	79	80	81	82	83	84	85	86	87	88
X_j	0	0	0	1	1	1	0	1	0	0	0
是否建立物流节点	否	否	否	是	是	是	否	是	否	否	否
a_j	879	880	881	882	883	884	885	886	887	888	889
j	89	90	91	92	93	94	95	96	97	98	99
X_j	1	0	1	0	0	0	0	0	0	1	1
是否建立物流节点	是	否	是	否	否	否	否	否	否	是	是
a_j	890	891	892	893	894	895	896	897	898	899	900
j	100	101	102	103	104	105	106	107	108	109	110
X_j	0	0	0	0	1	0	0	0	0	1	1
是否建立物流节点	否	否	否	否	是	否	否	否	否	是	是

也就是说，在编号为 791、794、795、806、807、808、815、818、824、825、841、844、851、858、865、867、871、872、873、875、879、881、888、889、894、899、900 的这 27 个交通货运区域中心处建立物流节点。如下图 4.3.1 所示为 matlab 求解所得的最优节点分布示意图：

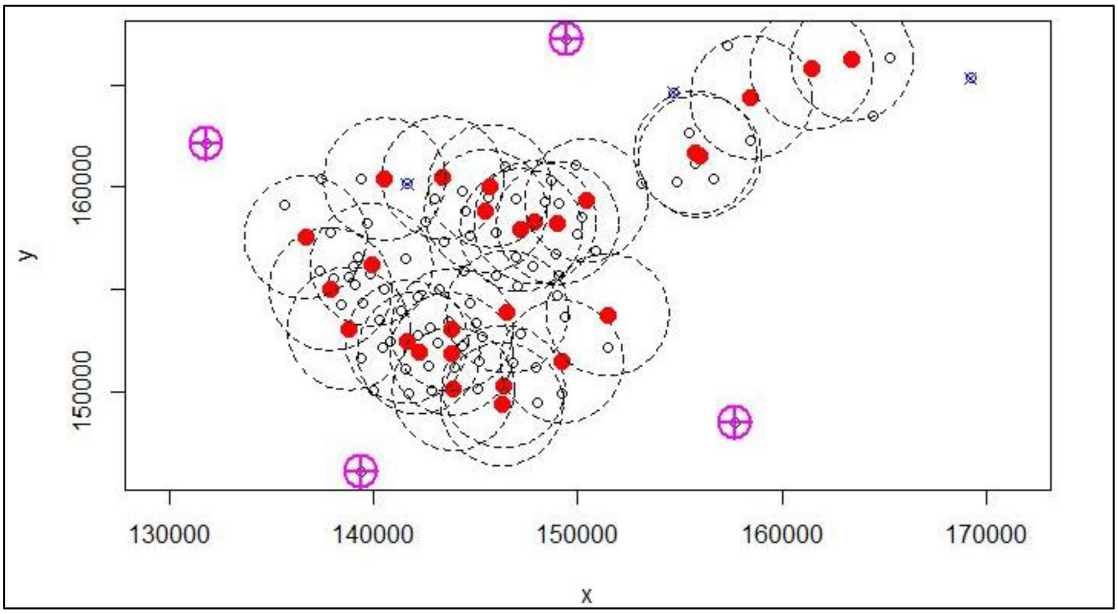

图 4.3.1 节点分布图

上图中四个紫色符号表示的是四个物流园区，红色圆点表示一级或二级节点位置，虚线圆圈表示的是各个节点的服务范围。图中用带叉的蓝色圆圈符号表示的是 866、895、896 三个交通货运区域中心，这三个交通货运区域的吞吐量原本就没有达到饱和，因此无需被节点服务范围覆盖。黑色空心圆圈和表示的是仙林区其余 107 个交通货运区域的中心，可以看出，这些需要地下物流系统分担运输压力以疏通交通拥堵的区域全部都被物流节点的服务范围覆盖。

Matlab 解出的 Y_{ij} 结果中非零的如下图所示：

表 4.3.2 各物流节点服务范围对各区域中心货运的分担关系

i	j	Y_{ij}	i	j	Y_{ij}	i	j	Y_{ij}
1	4	1	38	34	1	71	91	0.858750549
2	4	1	39	35	1	72	68	1
3	5	1	40	17	0.42491189	73	75	1
4	1	1	40	28	0.57508811	74	85	1
5	1	1	41	81	1	75	75	1
6	1	1	42	83	1	77	77	1
7	5	1	43	28	1	78	68	1
8	17	1	44	28	0.899910241	79	77	1
9	1	1	44	34	0.100089759	80	91	1
10	1	1	45	81	1	81	81	1
11	5	1	46	83	0.764161434	82	81	1
12	5	1	46	85	0.235838566	83	85	1
13	1	0.908413036	47	81	0.04297657	84	82	1
13	18	0.091586964	47	82	0.95702343	85	91	1
14	1	1	48	83	1	86	85	1
15	4	1	49	82	0.597238585	87	81	0.153182452
16	17	1	49	83	0.402761415	87	89	0.846817548

17	5	0.35404889	50	82	1	88	83	1
17	16	0.64595111	51	54	1	89	89	1
18	18	1	52	51	1	90	85	0.57699319
19	25	1	53	25	0.306608039	90	91	0.42300681
20	17	1	53	51	0.091801791	91	75	1
21	25	1	53	54	0.60159017	92	83	1
22	18	0.813804001	54	68	1	93	91	1
22	35	0.186195999	55	54	1	94	89	1
23	17	0.361260014	56	61	0.36314249	95	99	1
23	25	0.638739986	56	68	0.63685751	96	98	1
24	17	1	57	61	1	97	98	1
25	16	1	58	51	0.867732388	98	99	1
26	4	1	58	68	0.132267612	99	99	1
27	28	1	59	51	1	100	89	1
28	25	1	60	61	1	101	99	1
29	16	1	61	61	1	102	109	1
30	34	1	62	54	1	103	98	0.11236453
31	16	0.020963868	63	61	1	103	99	0.614161261
31	34	0.979036132	64	61	1	103	104	0.273474208
32	18	0.843054685	65	85	1	104	104	1
32	34	0.156945315	66	81	1	107	109	1
33	25	1	67	54	1	108	104	1
34	28	1	68	68	1	109	110	1
35	35	1	69	77	1	110	110	1
36	35	1	70	75	1			
37	51	1	71	75	0.141249451			

表中未列出的(i,j)组合得到的 Y_{ij} 为 0，即节点 a_j 不分担区域 b_i 的物流吞吐量。Yij=1 表示节点 a_j 分担区域 b_i 的全部额外物流吞吐量；0<Yij<1 则表示节点 a_j 与其他节点共同分担区域 b_i 的额外物流吞吐量。

4.4 计算各节点的实际货运量

各交通货运区域的地下货运量为其额外吞吐量（见表 4.1.1）。根据表 4.3.2 中各物流节点承担各交通货运区域额外吞吐量的比例可以求得各个节点的实际货运量，如下表所示：

表 4.4.1 各节点实际货运量

节点编号 j	区域中心编号	节点实际货运量（吨）
1	791	3000
2	794	1604.872
3	795	3000
4	806	3000
5	807	3000

6	808	3000
7	815	3000
8	818	3000
9	824	3000
10	825	3000
11	841	3000
12	844	2695.581
13	851	3000
14	858	3000
15	865	2265.476
16	867	2688.124
17	871	3000
18	872	3000
19	873	3000
20	875	3000
21	879	3000
22	881	3000
23	888	3000
24	889	3000
25	894	2909.762
26	899	2003.533
27	900	2073.626

4.5 确定一级节点的位置

在 4.3 中我们找出了节点的分布位置，但尚未对一级节点和二级节点进行划分。在这一节我们从经济节省的角度出发对一、二级节点进行划分，并确定一级节点所辖二级节点。

根据题目中的规定，一级节点与物流园区相连且一级节点之间连通；二级节点仅与本区域一级节点连通;；园区只与最近的一级节点连通。由于园区与仙林区各节点的距离远远大于相邻节点间的距离，而一级节点又需要与园区相连，显然，若一级节点数目多于 4 就会造成过多不必要的浪费，因此从成本上考虑，我们仅从与园区距离较近的节点中选择 4 个作为一级节点建设。

我们针对每个园区选择与其距离最近的三个节点作为备选的一级节点，从而得到 $3^4=81$ 种备选方案：

园区 1 附近：795，807，815

园区 2 附近：825，808，791

园区 3 附近：879，881，889

园区 4 附近：858，867，844

对于这些备选方案，一、二级节点数目相同，一级节点到园区的距离差距不大，可以从中选择一级节点转运率最低的方案作为最优方案。

根据定义，一级节点转运率（φ）为从物流园区经由最近的一级节点转运至其他所有一级节点的货物量占该物流园区总**出货量**的百分比。从园区发出货物的

运输渠道有三种：1）直接通过地上运输出货，我们将这一部分货运量称为**园区地上出货量**；2）通过地下物流系统发往最近的一级节点，或通过该一级节点发往其附属的二级节点，我们将这一部分货运量称为**园区一级节点非转运量**；3）通过地下物流系统经由最近的一级节点发往其他一级节点及其附属的二级节点，我们将这一部分货运量称为**园区一级节点转运量**。由定义可知，

$$\text{园区一级节点转运率 } \varphi=\frac{\text{园区一级节点转运量}}{\text{园区地上出货量}+\text{园区一级节点非转运量}+\text{园区一级节点转运量}}$$

$$\varphi=\frac{\text{园区发货量}-\text{园区地上发货量}-\text{园区一级节点非转运量}}{\text{园区总出货量}}。$$

园区吞吐总量可从现状全天 OD 表中获得：园区 1 为 37740.73 吨，园区 2 为 37149.64 吨，园区 3 为 36597.7 吨，园区 4 为 17377.48 吨，共计 128865.538 吨。

从现状全天 OD 表中可以得到各区域每天与四个园区之间的货物吞吐量，并求得它们每天与四个园区的货物吞吐量总和。现在我们已经满足了仙林区各个交通货运区域交通畅通，也就是说，我们可以将每个区的额外吞吐量交由地下物流系统进行运输，4.1 中得出的每个区的额外吞吐量为该区域可放入地下物流系统的地下运输额度。由于我们要满足进出四个园区的货物优先放入地下物流系统，且转运率要尽可能小，若某区域与四个园区的货物交换总量小于其地下运输额度，则该区域与园区的货物交换全部使用地下物流体系；若该区域与园区的总与货物交换量大于其地下运输额度，则该区域的全部地下运输额度用于运输与园区之间的货物交换，且优先满足园区出货、该区域收货这个方向的货物运输，其余未完成部分通过地面运输，且该区域与其它区域之间的货物交换全部通过地面运输。因此，一个区域与四个园区之间通过地下物流系统交换的货物总量为其地下运输额度和与园区货物交换总量中的较小值，即：

区域与园区的地下货运量=min(区域的地下运输额度，区域与园区货物交换总量)

表 4.5.1 园区和交通货运区域之间的货物运输

	…	888	889	890	891	…	交通货运区域总计
园区 1	…	77.182	126.495	275.981	281.78	…	37740.731
园区 2		130.926	187.445	477.626	333.729		37149.636
园区 3		257.693	98.731	2335.34	326.786		36597.696
园区 4		6.756	22.749	41.776	64.166		17377.475
园区总计		472.557	435.42	3130.723	1006.461		128865.538
地下运输额度		697.8439879	604.2784598	1342.454762	424.1167895		76194.57847
与园区间的地下运量		472.557	435.42	1342.454762	424.1167895		72307.06097

以交通货运区域 888、889、890 和 891 为例，由于区域 888 每日和四个园区的货物交换量仅有 472.557 吨，区域 889 每日和四个园区的货物交换量仅有 435.42 吨，尽管这两个区域放入地下物流系统运输的货物量分别为 697.8439879 吨和 604.2784598 吨，但其中与园区间的地下运量分别只有 472.557 吨和 435.42

吨；区域 890 每日和园区间的货物运输有 3130.723 吨，区域 891 每日和园区间的货物运输有 1006.461 吨，但由于它们的地下运输额度不足，因此与园区间的地下运输量与它们的地下运输额度相同，分别为 1342.454762 吨和 424.1167895 吨。

因为园区之间不存在货物运输，因此四个园区每天通过地下物流系统运输的货物总量即为所有交通货运区域与园区之间的地下物流系统运输量的总和，即 72307.06097 吨。园区每天通过地上运输的总吞吐量为园区总吞吐量减去园区地下吞吐量，即 128865.538-72307.06097=56558.477029736 吨。

我们不知道单个园区货物通过地上运输和地下运输的比例，因此以全部园区的地上运输和地下运输比例（72307.06097/128865.538≈56%）来进行估计，可以得到四个园区的地下吞吐量分别约为 21176.50 吨、20844.84 吨、20535.14 吨、9750.5832 吨。使用这个比例，我们也可以估计 4 各园区的地下发货量分别为其总发货量的 56%，即分别约为 10327.16 吨、10174.37 吨、10031.21 吨、4734.13 吨。

这样一来，在上面列举的 81 种备选方案中，每一种方案以就近原则将二级节点分配给四个一级节点，用园区的地下发货量减去最近的一级节点地下运输额度和附属于该一级节点的二级节点地下运输额与该园区发往该一级节点及其附属二级节点货物量之和的值取小，得到该园区的一级节点转运量，从而计算出该园区的一级节点转运率，即

$$\text{园区的一级节点转运率}=\frac{\text{该园区的一级节点转运量}}{\text{园区地下发货量}}$$

$$=\frac{\text{园区地下发货量}-\min(\text{对应一级节点及附属二级节点地下运输额度和，该园区发往相应一级节点及其附属二级节点的货物量和})}{\text{园区地下发货量}}$$

为了选出使得每个园区的一级节点转运率都较小的方案，我们对每一种方案的四个园区的一级节点转运率求几何平均值进行比较排序，得到最佳方案为分别选择 17、1、99、77 号节点作为园区 1、2、3、4 的一级节点，四个园区的一级节点转运率分别为 0.20583760，0.20557236，0.03184700，0.09319932。

也就是说，编号为 807、791、889、867 的交通货运区域中心处设置一级节点，园区 1 附近的一级节点在 807 区域中心，下辖 795、806、807、815、841、844 区域中心处的二级节点；园区 2 附近的一级节点在 791 区域中心处，下辖 791、794、808、824、825 区域中心的二级节点；园区 3 附近的一级节点在 889 区域中心处，下辖 872、879、888、889、894、899、900 区域中心处的二级节点；园区 4 附近的一级节点在 867 区域中心处，下辖 851、858、865、867、871、873、875、881 区域中心处的二级节点。一二级节点的分布如下图所示：

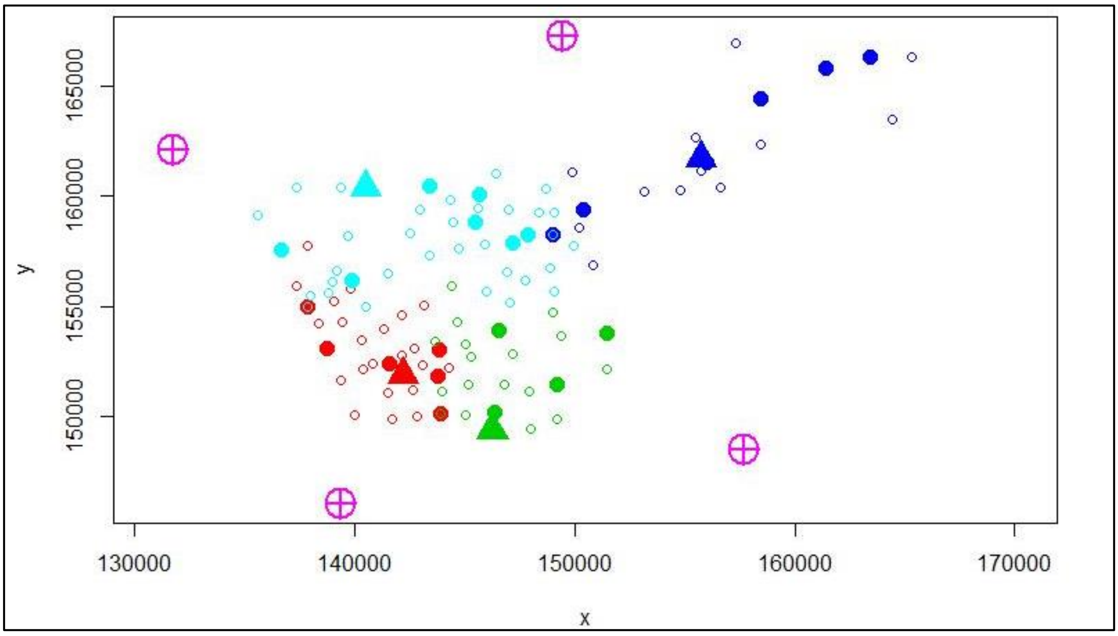

图 4.5.1 一级节点及所属二级节点分布图

图中紫色带十字的圆圈符号表示四个物流园区；三角形图案代表四个一级节点；与三角形颜色相同的实心圆圈代表相应的二级节点；空心圆圈代表其它交通货运区域中心。

4.6 问题一的结论

综上所述，应当在仙林地区设置一级节点 4 个，二级节点 23 个。各一级节点的位置、服务范围、所辖二级节点、实际货运量和转运率如下表所示：

表 4.6.1 四个一级节点信息表

临近园区	园区 1	园区 2	园区 3	园区 4
节点编号 j	17	1	99	77
区域中心编号	807	791	889	867
x 坐标	142199.9400	146277.0000	155757.6200	140522.9900
y 坐标	151897.7700	149377.0500	161698.6300	160396.7500
直接服务区域	798、806、810、813、814、830	794、795、796、799、800、803、804	885、888、889、891、893	859、867、869
所辖二级节点	795、806、815、818、841、844	794、808、824、825	872、879、888、894、899、900	851、858、865、871、873、875、881
节点实际货运量（吨）	3000	3000	3000	2688.124
一级节点转运率	0.20583760	0.20557236	0.09319932	0.03184700

各二级节点的位置、服务范围、实际货运量信息如下表所示：

表 4.6.2 二级节点信息表

区域中心编号	x 坐标	y 坐标	节点实际货运量（吨）	服务区域
794	146343.9800	150237.0900	1604.872	791、792、805、816
795	143903.2700	150127.8400	3000	793、797、801、802、807
806	143800.1000	151838.3900	3000	807、815、819、821
808	149214.8100	151472.2300	3000	803、808、812、822
815	141612.0200	152424.7800	3000	809、811、813、818、823、843
818	143839.8000	153020.8300	3000	817、824、830、833、834
824	146543.6800	153913.7800	3000	820、821、822、828、834
825	151449.4400	153764.2500	3000	812、825、826、829
841	138760.2800	153074.0100	3000	827、842、843、848、849
844	137872.6200	155016.2100	2695.581	841、843、845、852、857
851	139881.7800	156200.2800	3000	846、847、850、851、853、854
858	136656.2300	157553.5500	3000	844、846、848、858、862、868
865	143385.2200	160463.4800	2265.476	860、861、863、865、881
871	147221.0600	157904.4100	3000	831、835、837、856、871、872、877
872	149007.8800	158255.2300	3000	837、839、840、874
873	147900.6400	158283.6700	3000	832、836、838、839、878、882
875	145488.5900	158833.2500	3000	836、855、864、873、876、880

879	150413.2200	159364.6600	3000	877、879、884、890
881	145693.6500	160063.0700	3000	861、870、875、880、883
888	155973.6900	161520.8100	3000	886、887、893
894	158487.3400	164413.3800	2909.762	893、894、898
899	163428.0200	166293.3500	2003.533	892、897
900	161442.7300	165833.1700	2073.626	899、900

五、问题 2 的解答

5.1 节点的成本

由题意可知，一级、二级节点的建设成本分别约为 1.5 亿元/个、1 亿元/个。我们现需要建设一级节点 4 个、二级节点 23 个，其总成本为

$$1.5 \times 4 + 1 \times 23 = 29\text{（亿元）}$$

节点的设计年限为 100 年，因此节点的日均成本为

$$29 \times \frac{1}{365 \times 100} = 7.95 \times 10^{-4} \quad \text{（亿元）}$$

即 7.95 万元。

5.2 各类通道及其长度

我们将一个一级节点及其下辖的二级节点服务的总范围称为一个大型区域，一级节点 1、2、3、4 分别服务于大型区域 1、2、3、4。

首先，根据通道连接的对象，可以将通道分为三种：连接园区和一级节点的通道；连接两个一级节点的通道；各大型区域内连接起一级节点和各二级节点的通道。

根据通道中的车辆载重，可以将通道分为两种：使用载重 10 吨的地下运输车辆的通道和使用载重 5 吨的地下运输车辆的通道。

根据通道中的轨道数目，可以将通道分为两种：双向双轨通道和双向四轨通道。

5.2.1 园区到一级节点的隧道

由“各区域中心点及面积”表格中的数据可以求得各点之间的距离。4 个园区及 4 个一级节点的坐标分别为：

表 5.2.1.1 园区和一级节点坐标

	x（米）	y（米）
园区 1	139382.3437	146047.0951
园区 2	157688.4185	148489.0310
园区 3	149441.1622	167278.6394

园区 4	131764.7130	162103.0925
一级节点 1	142199.9400	151897.7700
一级节点 2	146277.0000	149377.0500
一级节点 3	155757.6200	161698.6300
一级节点 4	140522.9900	160396.7500

由此可求得各园区与最临近的一级节点之间的距离：

园区 1 与一级节点 1： 6493.7825 米

园区 2 与一级节点 2：11445.9185 米

园区 3 与一级节点 3： 8428.1756 米

园区 4 与一级节点 4： 8922.9491 米

因车辆加速度 1 米/秒 ^{2}，运行速度 13.5 米/秒，车辆加速、减速共需要 27 秒，变速期间行驶 182.25 米，忽略车辆长度可以求得这四条线路的单程时间 t 分别约为：8.2 分钟、14.3 分钟、10.6 分钟、11.2 分钟。由于地下物流系统每天运营时间为 18 小时，每 12 分钟可发车 1 班，因此，各线路每天单向最多发车班次为 $\left\lfloor\frac{18\times60-t}{12}\right\rfloor+1$班，即分别为 90、89、90、90 班。

每班列车由 4 至 8 辆运输车辆构成，每辆车载重为 10 吨，因此每班运载车辆的载重为 40-80 吨之间。双向双轨通道的单向运载量分别为 3600-7200 吨、3560-7120 吨、3600-7200 吨、3600-7200 吨；双向四轨列车的单向运载量分别为 7200-14400 吨、7120-14240 吨、7200-14400 吨、7200-14400 吨。

表 5.2.1.2 园区到一级节点的隧道

	园区 1——一级节点 1	**园区 2——一级节点 2**	**园区 3——一级节点 3**	**园区 4——一级节点 4**
距离（米）	6493.785159	11445.91848	8428.1756	8922.9491
单趟时长（分钟）	8.242018715	14.35576356	10.63015506	11.24098654
每日单向车次	90	89	90	90
双向双轨(吨）	3600~7200	3560~7120	3600~7200	3600~7200
双向四轨（吨）	7200~14400	7120~14240	7200~14400	7200~14400

在 4.5 节中我们已经算出了四个园区通过地下物流系统运输的总和 72307.06 吨，并通过地下运输总和占园区吞吐量总和 128865.54 吨的比例 56%来估计了四个园区的地下吞吐量分别约为 21176.50 吨、20844.84 吨、20535.14 吨、9750.58 吨。

从表 4.5.1 园区和交通货运区域之间的货物运输中可以看出，园区和每个交通货运区域之间的货物运输量的一半都 6 小于园区和该交通货运区域的进、出货物量中的最小值。由于同一隧道中两个方向的轨道数应当相同，所以我们可以安排所有区域与园区通过地下接收和发送的货物量相等。这样，四个园区的地下收货量（或地下发货量）分别约为 10588.25 吨、10422.42 吨、10267.57 吨和 4875.29 吨。

这样，园区 1、2、3、4 到节点 1、2、3、4 的隧道单向日运输量不应低于 10588.25 吨、10422.42 吨、10267.57 吨和 4875.29 吨。参照表 5.2.1.2 可知，园区 1 到一级节点 1、园区 2 到一级节点 2、园区 3 到一级节点 3 均应选用双向四轨隧道，园区 4 到一级节点 4 应当选用双向双轨轨道。

用 T_1 表示四个园区至最近一级节点的通道长度之和，用 T_{11}、T_{12}、T_{13}、T_{14} 分别表示园区 1、2、3、4 至一级节点 1、2、3、4 之间的通道长度。所以 T_{11}=6.49km，T_{12}=11.45km，T_{13}=8.43km，T_{14}=8.92km，T_1=35.29km.

载重 10 吨的车辆的双向双轨隧道造价为每千米 4 亿元，双向四轨隧道为每千米 5 亿元，因此园区到一级节点的隧道总成本为：

$5\times(6.49+11.45+8.43)+4\times 8.92=167.53$ （亿元）

隧道的设计年限为 100 年，因此园区到一级节点的隧道日均成本为：

$$167.53\times\frac{1}{365\times 100}=4.590\times 10^{-3}\quad（亿元）$$

即 45.90 万元。

5.2.2 大型区域内的隧道

根据题中的规定，一级节点及下属二级节点之间全部使用载重为 5 吨的运载车辆。由于一趟车可以有 4-8 辆运载车，因此每趟车的运载量为 20-40 吨之间。区域间的距离都很近，一趟车的运行时间小于发车间隔 12 分钟，因此一天 18 小时的运营时间中单向最多可以有 90 趟车，双向双轨隧道的日单向流量为 1800-3600 吨之间；双向四轨隧道的日单向流量为 3600-7200 吨之间。

确定大型区域内的隧道可分为两步进行：

1）**选择连接方案：**确定一个一级节点与其下属的各二级节点之间的最优连接方式。一级节点无须直接与每个下属二级节点相连，可以通过其下属的其他二级节点连接；二级节点仅能与其所属的一级节点或本区其它二级节点直接相连。我们采用分步优化的做法，首先在每一个一级节点辖区内选择最为经济的连接方案相当于在既定的点中寻找一条能够连接所有点且总长度最短的线路，这是一个以该区域一级节点为根的最小生成树问题，我们采用 Prim 算法求解该问题。

2）**确定隧道轨道数：**然后根据最优连接方式对每条线路的流量要求来选择合适的隧道，即为“限定隧道，再定轨道数”的做法。

我们使用 R 语言对该问题进行计算，得到了分别适合四个大型区域的隧道连接方案：

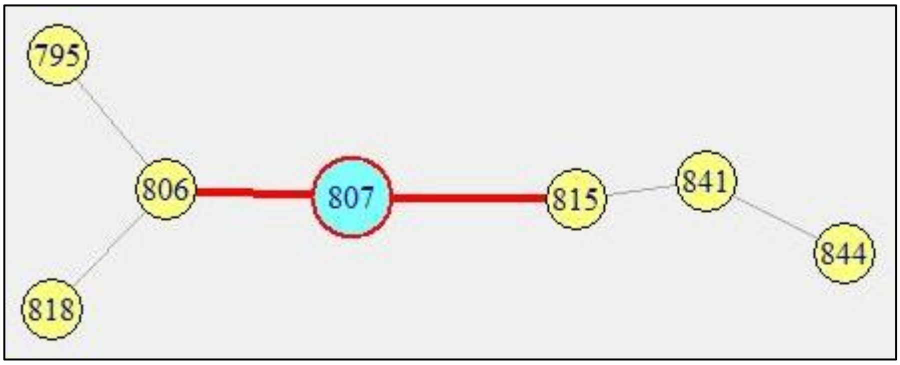

图 5.2.2.1 大型区域 1 内的隧道示意图

图5.2.2.1为一级节点1服务的大型区域1内的一二级节点间隧道连接方案。区域 807 的中心为一级节点 1 所在位置，其余圆圈中的是从属于该一级节点的二级节点。由于二级节点只能与本区域的节点相连，因此位于区域 795、818、844 均为地下物流系统的端点。在 5.2.1 节中已经说明，我们可以安排所有区域与园区通过地下接收和发送的货物量相等，即线路的单向货运量为线路总货运量的

1/2。因此 795-806、806-818 和 841-844 间的单向线路货运量分别等于节点 795、818 和 844 的实际双向货运量的一半，参照表 4.6.2 可知，分别为 1500 吨、1500 吨、1348 吨。

线路 806-807 的货运量为 795-806、818-806 的货运量和节点 808 的实际货运量之和减去节点 795、818 分别与 806 之间的地下货运量。由于地下货运量额度大部分被用于满足区域与园区之间的货物交换，区域间的货物交换大部分通过地上进行，只有区域的额外吞吐量大于该区域与四个园区的货物交换量之和时才会有区域间的货物通过地下物流系统运送，因此区域间的地下物流交换量很少。况且此处我们需要计算线路单向流量的最大值，因此可以不考虑减去区域间的地下物流量，这样，线路 806-807 的单向货运量约为$1500+1500+\frac{1}{2}\times3000=4500$吨；同理可得，线路 815-841 的单向货运量约为$1348+\frac{1}{2}\times3000=2848$吨，线路 807-815 的单向货运量约为$2848+\frac{1}{2}\times3000=4348$吨。

因此，在大型区域 1 内，线路 795-806、818-806、815-841、841-844 仅需要双向双轨隧道，线路 806-807、807-815 需要双向四轨隧道。

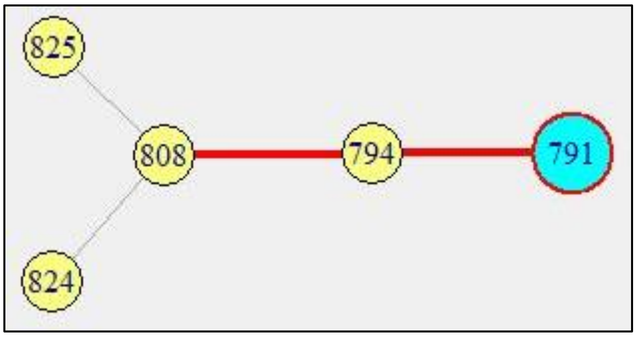

图 5.2.2.2 大型区域 2 内的隧道示意图

图 5.2.2.2 为大型区域 2 内的一二级节点间隧道连接方案。区域 791 的中心为一级节点 2 所在位置，其余圆圈中的是从属于该一级节点的二级节点。按照同样的方法可以算出，线路 791-794、794-808、808-824、808-825 的日单向货运量分别为为 5300 吨、4500 吨、1500 吨、1500 吨

因此，在一级节点 2 的服务区域内，线路 808-824、808-825 仅需要双向双轨隧道，线路 791-794、794-808 需要双向四轨隧道。

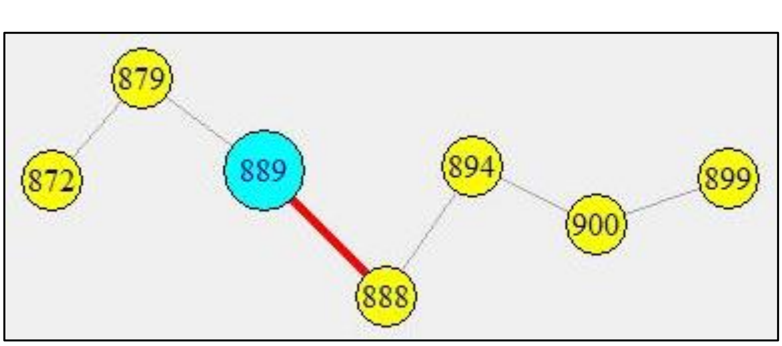

图 5.2.2.3 大型区域 3 内的隧道示意图

图 5.2.2.3 为大型区域 3 内的隧道连接方案。区域 889 的中心为一级节点 3 所在位置，其余圆圈中的是从属于该一级节点的二级节点。按照同样的方法可以算出，线路 872-879、879-889、888-889、888-894、894-900、899-900 的日单向货运量分别为为 1500 吨、3000 吨、4993 吨、3494 吨、2039 吨、1001 吨。

因此，在大型区域 3 内，线路 872-879、879-889、888-894、894-900、899-900 仅需要双向双轨隧道，线路 888-889 需要双向四轨隧道。

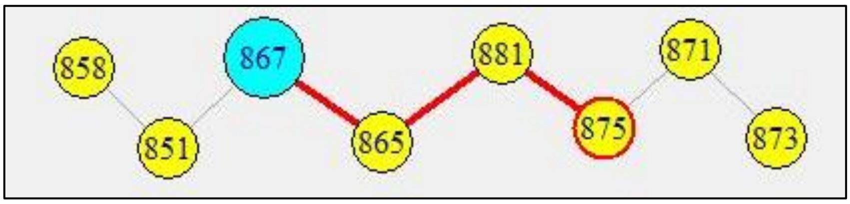

图 5.2.2.4 一大型区域 4 内的隧道示意图

图 5.2.2.4 为大型区域 4 内的一二级节点间隧道连接方案。区域 867 的中心为一级节点 4 所在位置，其余圆圈中的是从属于该一级节点的二级节点。按照同样的方法可以算出，线路 851-858、851-867、865-867、865-881、875-881、871-875、871-873 的日单向货运量分别为 1500 吨、3000 吨、7133 吨、6000 吨、4500 吨、3000 吨、1500 吨。

因此，在大型区域 4 内，线路 851-858、851-867、871-875、871-87 仅需要双向双轨隧道，线路 865-867、865-881、875-881 需要双向四轨隧道。

通过《各区域中心点及面积》表中的数据我们可以计算各个线路的长度。大型区域内连接一二级节点的线路、日单向货运量及隧道类型如下表所示：

表 5.2.2.1 大型区域内的线路信息表

一级节点	线路	线路长度（千米）	日单向货运量（吨）	隧道类型
1	795-806	1.71	1500	双向双轨
	806-818	1.18	1500	双向双轨
	841-844	2.14	1348	双向双轨
	806-807	**1.60**	**4500**	**双向四轨**
	815-841	2.92	2848	双向双轨
	807-815	**0.79**	**4348**	**双向四轨**
2	**791-794**	**0.86**	**5300**	**双向四轨**
	794-808	**3.13**	**4500**	**双向四轨**
	808-824	3.62	1500	双向双轨
	808-825	3.20	1500	双向双轨
3	872-879	1.79	1500	双向双轨
	879-889	5.83	3000	双向双轨
	888-889	**0.28**	**4993**	**双向四轨**
	888-894	3.83	3494	双向双轨
	894-900	3.28	2039	双向双轨
	899-900	2.04	1001	双向双轨

4	851-858	3.50	1500	双向双轨
	851-867	4.25	3000	双向双轨
	865-867	**2.86**	**7133**	**双向四轨**
	865-881	**2.34**	**6000**	**双向四轨**
	875-881	**1.25**	**4500**	**双向四轨**
	871-875	1.97	3000	双向双轨
	871-873	0.78	1500	双向双轨

一级节点及附属二级节点的线路长度总和为 55.15km，其中双向四轨轨道 13.11km，双向双轨轨道 42.04km。已知载重 5 吨车辆的隧道双向四轨造价 3.5 亿元/千米，双向双轨造价 3 亿元/千米，所以大型区域内的线路总成本为：

$$3 \times 42.04 + 3.5 \times 13.11 = 172.00\text{（亿元）}$$

由于通道的设计年限为 100 年，则大型区域内的线路日均成本为：

$$172.00 \times \frac{1}{365 \times 100} = 4.712 \times 10^{-3} \quad\text{（亿元）}$$

即 47.12 万元。

5.2.3 一级节点之间的隧道

需要通过一级节点间隧道运输的货物包含两部分：从园区到其他大型区域的和从大型区域到其他大型区域的。

从园区到其他大型区域的货物量可以通过园区的一级节点转运率来计算：

$$\text{园区发往其他大型区域的货物量} = \text{园区总发货量} \times \text{园区的一级节点转运率}$$

通过对《现状全天 OD》中园区 1、2、3、4 每天发往其它大型区域的发货量进行统计，可以根据各个园区的一级节点转运率估算各个园区通过一级节点间隧道发往其他大型区域的货物量：

表 5.2.3.1 园区向其他大型区域的发货量地下部分

	大型区域 1 地下收货	大型区域 2 地下收货	大型区域 3 地下收货	大型区域 4 地下收货
园区 1 发货	-	870.27047	720.8151	964.7102
园区 2 发货	713.00449	-	991.7935	844.2121
园区 3 发货	88.82185	58.23348	-	150.4225
园区 4 发货	106.81415	124.07709	188.1442	-

大型区域之间的货运交换比较复杂，我们可以通过区域间货物通过地下物流系统的交换量和区域间货物的总交换量比例来进行估算。

由表 4.5.1 可知，仙林区 110 个交通货运区域每天的地下吞吐量总和为 76194.58 吨，我们用 110 个交通货运区域每天的地下吞吐量总和（也是四个大型区域的地下吞吐量总和）减去每日园区通过地下发往四个大型区域的货物量，可以得到四个大型区域每天通过地下发给园区的货物量和四个大型区域每天相互通过地下交换的货物量之和。将这一差值除以四个大型区域发往园区和其它区域的总和（相当于四个大型区域每天发给园区的货物量和四个大型区域每天交换的货物量之和），可以得到区域间货物及区域发给园区货物之和通过地下物流系统的交换量和区域间货物及区域发给园区货物的总交换量比例，约 15.86%。具体

的数学表达式及注释如下：

大型区域甲及园区甲发给大型区域乙及园区乙中通过节点间隧道运输的货物量 A

$= $ 园区甲发给大型区域乙的总货物量 B $\times$ 节点甲的转运率 C

$+$ 节点甲及附属节点发给大型区域乙及园区乙的总货物量 D

$\times$

节点甲及附属节点发给大型区域乙及园区乙的总货物量中通过节点间隧道运输的比例 E

其中 A 是我们想要估计的量，B 和 D 可以从《现状全天 OD》中获取，C 在第一题中已经计算过，假设对于任意（节点，大型区域）对，E 的值大致相同，那么可以通过估计所有节点发给所有大型区域的总货物量中通过节点间隧道运输的比例来估计 E，具体表达式如下：

$$E = \frac{\text{所有节点发往所有大型区域及对应园区的地下货运量}}{\text{所有节点发往所有大型区域及对应园区总货运量}}$$

$$= \frac{\text{所有节点地下货运额度} - \text{所有园区发往所有节点地下货运量}}{\text{所有节点间相互收发量} + \text{所有节点发往所有园区货运量}}$$

$$= \frac{76194 - 54798}{69082 + 65889} = 15.86\%$$

通过对《现状全天 OD》中属于各大型区域的交通货运区域每天发往其它大型区域的发货量进行统计，可以根据这一比例估算各个大型区域通过一级节点间隧道发往其他大型区域和园区的货物量：

表 5.2.3.2 大型区域向其他大型区域和园区的发货量地下部分

	园区 1 和大型区域 1 地下收货	园区 2 和大型区域 2 地下收货	园区 3 和大型区域 3 地下收货	园区 4 和大型区域 4 地下收货
大型区域 1 发	0	768.7028	635.675	888.6946
大型区域 2 发	950.7932	0	606.8334	507.0781
大型区域 3 发	925.6946	1217.258	0	653.265
大型区域 4 发	1610.631	976.7115	970.9674	0

由于园区和园区之间不存在货物运输，因此叠加上面两表可以获得大型区域和相应的园区构成的整体之间的地下货物交换量状况：

表 5.2.3.3 一级节点间通道的货运量

	园区 1 和大型区域 1 地下收货	园区 2 和大型区域 2 地下收货	园区 3 和大型区域 3 地下收货	园区 4 和大型区域 4 地下收货
园区 1 和大型区域 1 地下发货	0	1638.973	1356.49	1853.405

园区 2 和大型区域 2 地下发货	1663.798	0	1598.627	1351.29
园区 3 和大型区域 3 地下发货	1014.516	1275.491	0	803.6874
园区 4 和大型区域 4 地下发货	1717.445	1100.789	1159.112	0

现在我们可以规划一级节点之间的隧道线路。4 个一级节点之间存在 6 条可能的连接线路，如图 5.2.3.1 所示：

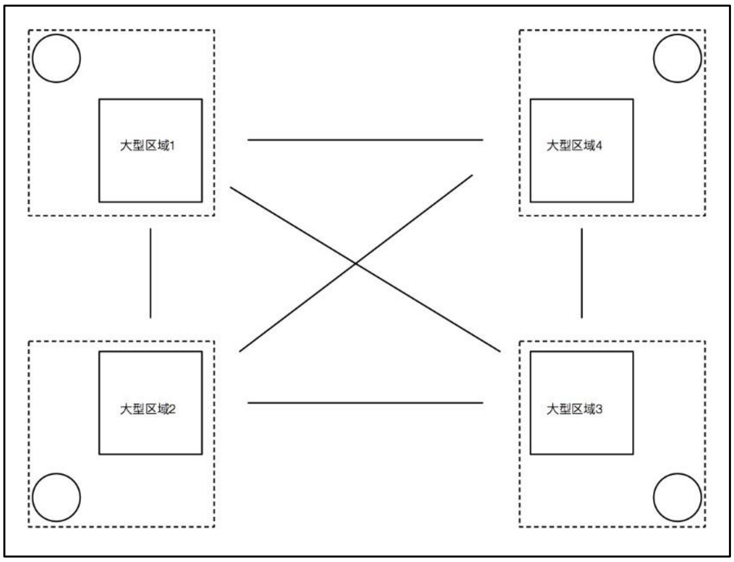

图 5.2.3.1 一级节点间可能的 6 条通道

对于一条备选的线路而言，我们可以计算该线路的造价和有、无该线路时的的运输成本差异。对于这条备选线路，根据其两端的地下物流货运量可以确定其单位距离的日均成本，再乘以线路长度可以得到线路的日均建造成本。如果修建该线路，则需要经该线路运输的货物在该线路上的运输成本为货物质量、单位质量单位里程运输价格和线路长度的乘积。如果不修建该线路，则需要经过其他线路绕行（在其他线路流量允许的情况下），运输成本为货物质量、单位质量单位里程运输价格和绕行线路长度的乘积。也就是说修建某线路后的日均节约成本为：

$$\text{日均节约成本} = \text{单位距离日均成本} \times \text{当前线路长度} - \text{货物质量} \times (\text{绕行线路长度} - \text{当前线路长度}) \times 1\, {}^{\text{元}}\!/\!_{\text{吨·公里}}$$

经过计算发现，这可能的 6 条线路的日均节约成本均为负，也就是说，与增修线路相比，绕行更节约成本。

4 个一级节点之间的距离如下表所示：

表 5.2.3.4 一级节点之间的距离

（单位：千米）

	一级节点 1 (807)	**一级节点 2 (791)**	**一级节点 3 (889)**	**一级节点 4 (867)**

一级节点 1 (807)	0	4.79	16.73	8.66
一级节点 2 (791)	4.79	0	15.55	12.43
一级节点 3 (889)	16.73	15.55	0	15.29
一级节点 4 (867)	8.66	12.43	15.29	0

由于要连接 4 个一级节点，至少需要 3 条通道，所以应当将四个节点连成线状，且通道总长度尽可能短、转运成本尽可能低。从通道长度最短的角度应该选取 3-4、4-1、1-2 三条通道，但这种通道仅比 4-1、1-2、2-3 通道节省了 0.26 千米的通道长度，但转运成本增大较多，因而总成本更高，因此最佳方案为将通道建在一级节点 4-1、1-2、2-3 之间。在这种方案下，通道 4-1 承载了 4-1、4-2、4-3 方向的地下物流量，1-2 承载了 1-2、1-3、4-2、4-3 的地下物流量，2-3 承载了 2-3、4-3、1-3 方向的地下物流量；通道 1-4 承载了 1-4、2-4、3-4 方向的地下物流量，2-1 承载了 2-1、3-1、2-4、3-4 方向的地下物流量，通道 2-3 承载了 2-3、1-3、4-3 方向的地下物流量。根据表 5.2.3.3 可以计算出，三条一级节点间通道两个方向的日运载量如下表所示：

表 5.2.3.5 一级节点之间隧道的流量

	4-->1（吨）	**1-->2（吨）**	**2-->3（吨）**
正向	3977.346	5255.364	4114.229
反向	4008.3824	4833.2916	3093.694

可以看出，各条线路的单向日流量均不超过双向四轨隧道的最大日流量 7200 吨，因此该方案可行，且三条通道均应选用双向四轨隧道。

一级节点间线路的总建设成本为：

$$(8.66 + 4.79 + 15.55) \times 3.5 = 101.5 \quad （亿元）$$

由于通道的设计年限为 100 年，因此一级节点间线路的的日均成本为

$$101.5 \times \frac{1}{365 \times 100} = 2.781 \times 10^{-3} \quad （亿元）$$

即一级节点间的线路的日均成本为 27.81 万元。

5.3 货物运输成本

由于在各条线路上货物运输成本均为 1 元/吨·公里，因此货物运输成本主要由货物重量和运输距离决定。

从园区到一级节点的货物运输成本为：

$$21176.50 \times 6.49 \times 1 + 20844.84 \times 11.45 \times 1 + 20535.14 \times 8.43 \times 1 + 9750.58 \times 8.92 \times 1 = 636195.31 \quad （元）$$

即 63.62 万元。

参见表 5.2.2.1，大型区域内的货物运输成本为各段线路长度与日单向货运量的两倍，为 327748 元，即 32.77 万元。

一级节点间的货物运输成本为：

$$(3977.346+4008.3824)\times 8.66\times 1+(5255.364+4833.2916)\times 4.79\times 1+(4114.229+3093.694)\times 15.55\times 1=229564.27 \quad （元）$$

即 22.96 万元。

三类通道上的货物运输成本共计 119.35 万元。

5.4 问题二结论

符合要求的节点位置及实际货运量如下表所示：

表 5.4.1 一级、二级节点的位置及实际货运量

节点编号		对应区域中心编号	x（米）	y（米）	实际货运量（吨）
一级节点 1		**807**	**142199.9400**	**151897.7700**	**3000**
下辖二级节点	二级节点 2	795	143903.2700	150127.8400	3000
	二级节点 3	806	143800.1000	151838.3900	3000
	二级节点 5	815	141612.0200	152424.7800	3000
	二级节点 6	818	143839.8000	153020.8300	3000
	二级节点 9	841	138760.2800	153074.0100	3000
	二级节点 10	844	137872.6200	155016.2100	2695.581
一级节点 2		**791**	**146277.0000**	**149377.0500**	**3000**
下辖二级节点	二级节点 1	794	146343.9800	150237.0900	1604.872
	二级节点 4	808	149214.8100	151472.2300	3000
	二级节点 7	824	146543.6800	153913.7800	3000
	二级节点 8	825	151449.4400	153764.2500	3000
一级节点 3		**889**	**155757.6200**	**161698.6300**	**3000**
下辖二级节点	二级节点 11	851	139881.7800	156200.2800	3000
	二级节点 12	858	136656.2300	157553.5500	3000
	二级节点 13	865	143385.2200	160463.4800	2265.476
	二级节点 14	871	147221.0600	157904.4100	3000
	二级节点 16	873	147900.6400	158283.6700	3000
	二级节点 17	875	145488.5900	158833.2500	3000
	二级节点 19	881	145693.6500	160063.0700	3000
一级节点 4		**867**	**140522.9900**	**160396.7500**	**2688.124**
下辖二级节点	二级节点 15	872	149007.8800	158255.2300	3000
	二级节点 18	879	150413.2200	159364.6600	3000
	二级节点 20	888	155973.6900	161520.8100	3000
	二级节点 21	894	158487.3400	164413.3800	2909.762
	二级节点 22	899	163428.0200	166293.3500	2003.533
	二级节点 23	900	161442.7300	165833.1700	2073.626

三种通道位置及其实际流量如下表所示：

表 5.4.2 各类通道的信息及实际货运量

通道种类	通道位置	车辆载重	通道类型	线路长度（千米）	通道实际流量（吨）
园区至一级节点通道	园区 1——一级节点 1(807)	10 吨	双向四轨	6.49	21176.5
	园区 2——一级节点 2(791)	10 吨	双向四轨	11.45	20844.84
	园区 3——一级节点 3(889)	10 吨	双向四轨	8.43	20535.14
	园区 4——一级节点 4(867)	10 吨	双向双轨	8.92	9750.58
大型区域内通道	795-806	5 吨	双向双轨	1.71	3000
	806-818	5 吨	双向双轨	1.18	3000
	841-844	5 吨	双向双轨	2.14	2696
	806-807	5 吨	双向四轨	1.6	9000
	815-841	5 吨	双向双轨	2.92	5696
	807-815	5 吨	双向四轨	0.79	8696
	791-794	5 吨	双向四轨	0.86	10600
	794-808	5 吨	双向四轨	3.13	9000
	808-824	5 吨	双向双轨	3.62	3000
	808-825	5 吨	双向双轨	3.2	3000
	872-879	5 吨	双向双轨	1.79	3000
	879-889	5 吨	双向双轨	5.83	6000
	888-889	5 吨	双向四轨	0.28	9986
	888-894	5 吨	双向双轨	3.83	6988
	894-900	5 吨	双向双轨	3.28	4078
	899-900	5 吨	双向双轨	2.04	2002
	851-858	5 吨	双向双轨	3.5	3000
	851-867	5 吨	双向双轨	4.25	6000
	865-867	5 吨	双向四轨	2.86	14266
	865-881	5 吨	双向四轨	2.34	12000
	875-881	5 吨	双向四轨	1.25	9000
	871-875	5 吨	双向双轨	1.97	6000
	871-873	5 吨	双向双轨	0.78	3000
一级节点间通道	4(867)-1(807)	5 吨	双向四轨	8.66	7985.73
	1(807)-2(791)	5 吨	双向四轨	4.79	10088.66
	2(791)-3(889)	5 吨	双向四轨	15.55	7207.923

各级节点和通道的分布如下图所示：

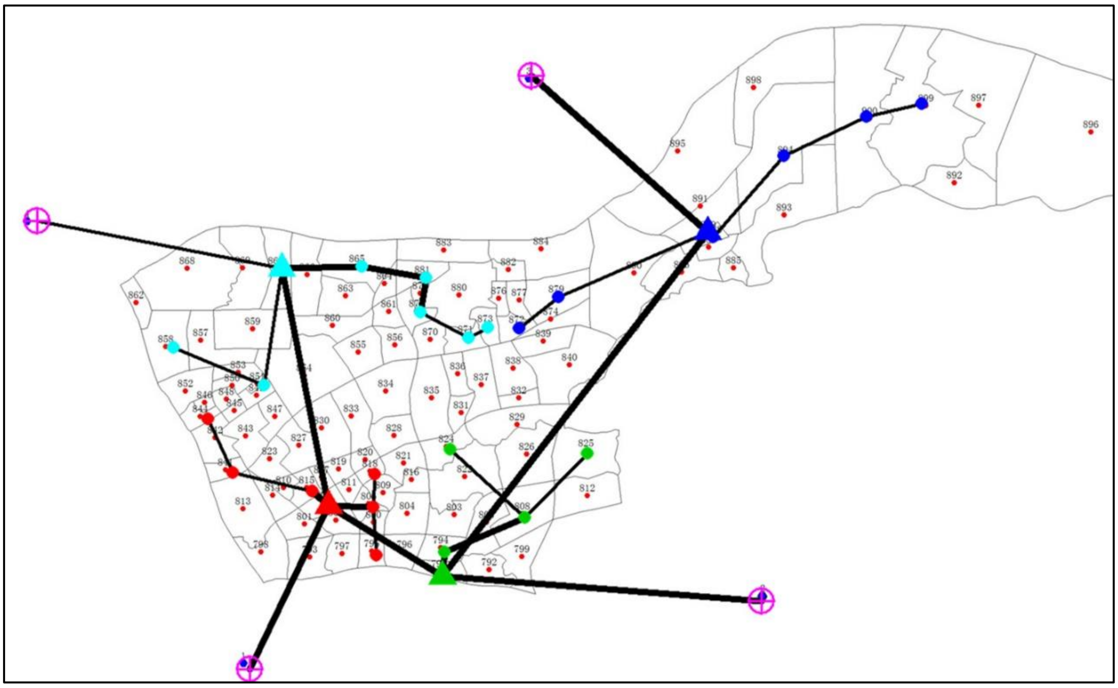

图 5.4.1 节点和通道的分布图

该方案的日均成本总计 248.13 万元，具体项目如下表所示：

表 5.4.3 方案日均成本说明

<table>
<tr><th colspan="4">最优情况造价费用名目</th><th>具体项目</th><th>日均成本（万元）</th><th>说明</th></tr>
<tr><td rowspan="7">总费用合计</td><td rowspan="7">248.13 万元/日</td><td rowspan="4">建造成本</td><td rowspan="4">128.78 万元/日</td><td>节点</td><td>7.95</td><td>一级节点 4 个，二级节点 23 个</td></tr>
<tr><td>园区至一级节点隧道</td><td>45.90</td><td>3 条隧道均采用 10 吨双向四轨，1 条采用 10 吨双向双轨</td></tr>
<tr><td>一级节点间隧道</td><td>27.81</td><td>共(1-4,1-2,2-3)三条 5 吨双向四轨隧道，总长度 29km</td></tr>
<tr><td>大型区域内通道</td><td>47.12</td><td>共 23 条隧道，总长度 55.15km，其中 5 吨双向双轨 15 条 42km，5 吨双向四轨 8 条 13.11km</td></tr>
<tr><td rowspan="3">运输成本</td><td rowspan="3">119.35 万元/日</td><td>园区到一级节点流量</td><td>63.62</td><td>4 个园区日均流量分别为 2.12,2.08,2.05,0.98 万吨，合计 7.23 万吨，或 63.6 万吨*km</td></tr>
<tr><td>一二级节点间流量</td><td>32.77</td><td>共计 32.77 万吨*km</td></tr>
<tr><td>一级节点间流量</td><td>22.96</td><td>共计 22.96 万吨*km</td></tr>
</table>

六、问题 3 的解答

6.1 降低成本的优化方案

在问题一中我们以最少的节点数目来完成疏通仙林地区交通拥堵状况的任务，并从中以使得园区的一级节点转运率最低为目标来从各个节点中选择一级节点。

在问题二中我们不仅要考虑节点修建成本，还要考虑通道的修建成本和货物的运输成本，并在问题一选定的一、二级节点基础上考虑线路规划问题。

从降低成本的角度考虑，问题一中节点数目尽可能少的目标依然存在，但我们无须保证一级节点转运率最小。因此使各个园区一级节点转运率最小这一限制可以解除，也就是说，我们还是基于问题一中选定的节点位置进行优化，但可以对节点中一级节点的选取标准进行更改。

当前，地下物流系统的日均成本由两个部分构成：日均建设成本和货物运输成本。日均建设成本包含日均节点建设成本、日均大型区域内通道建设成本、日均园区至最近一级节点通道建设成本和日均一级节点间通道建设成本。由于节点数目不会发生变化，日均节点建设成本没有改进空间。园区至最近一级节点通道成本与一级节点的选择高度相关，同时一级节点间通道的建设成本取决于一级节点的相对位置和它们之间的货运流量，这是由我们对一级节点的选择方案决定的。此外由于大型区域内的通道选择是通过寻找其最小生成树的方法来完成的，最小生成树是由几个点的空间分布决定的，而大型区域的构成则是由一级节点的选择确定的，因此，日均大型区域内通道的建设成本也取决于一级节点的选择。货物运输成本取决于线路排布，因此也是由一级节点的选取决定的。

综上所述，要从全局上优化地下物流系统，应当基于问题一中确定的 27 个节点位置，重新选择使得总成本最低的一级节点。

在问题一中我们以距离各个园区最近的 3 个节点作为该园区对应的一级节点的备选项，并从这$3^4 = 81$种方案中选出一级节点转运率最小的方案作为最佳方案。在本问题中，从全局上进行考虑，为了尽可能节约成本，我们以距离各个园区最近的 5 个节点作为该园区对应的一级节点备选点进行遍历，并从这$5^4 = 625$种方案中选出使得日均总成本最低的方案。

对于每组确定的一级节点，根据距离最近原则分配其相应的二级节点；对于每组一级节点和二级节点，用求解最小生成树算法求大型区域内最短的连接通道。考虑到路线单向流量不能超过双向四轨的最大流量 7200 吨/日，共有 66 种可行的方案。对于每种节点及轨道选择，分别根据每段路的流量进行双向双轨和双向四轨设计。这样，对于每组解我们可以计算其建造成本和运输成本，从而求得其日均总成本。如下表所示为这 66 中方案中日均总成本最低的五种方案：

表 6.1.1 成本最低的五种地下物流网络设计方案

优选方案	一级节点	下辖二级节点	日均总成本（万元）

1	806	791、794、795、806、807、815、818、824	228.6081
	825	808、825、871、872、873、879	
	889	888、889、894、899、900	
	851	841、844、851、858、865、867、875、881	
2	806	791、794、795、806、807、815、818、824	229.2412
	825	808、825、871、872、873、879	
	888	888、889、894、899、900	
	851	841、844、851、858、865、867、875、881	
3	806	791、794、795、806、807、815、818、824、875、881	229.288
	825	808、825、871、872、873、879	
	889	888、889、894、899、900	
	844	841、844、851、858、865、867	
4	806	791、794、795、806、807、815、818、824、875、881	229.9205
	825	808、825、871、872、873、879	
	888	888、889、894、899、900	
	844	841、844、851、858、865、867	
5	806	791、794、795、806、807、815、818、824、841	229.9858
	825	808、825、871、872、873、879	
	889	888、889、894、899、900	
	867	844、851、858、865、867、875、881	

在满足通行需求的基础上，我们选择总成本最低的方案 1 作为最佳方案。如下图所示为方案 1 的节点分布和通道分布，其中用紫色带十字圆圈表示的是四个园区，三角形符号表示四个一级节点，与各个三角形颜色相同的圆点表示从属于该一级节点的二级节点。连接各个节点或园区的黑色实线表示地下物流通道，其中加粗的实线条表示使用双向四轨隧道处，其余通道使用双向双轨隧道。

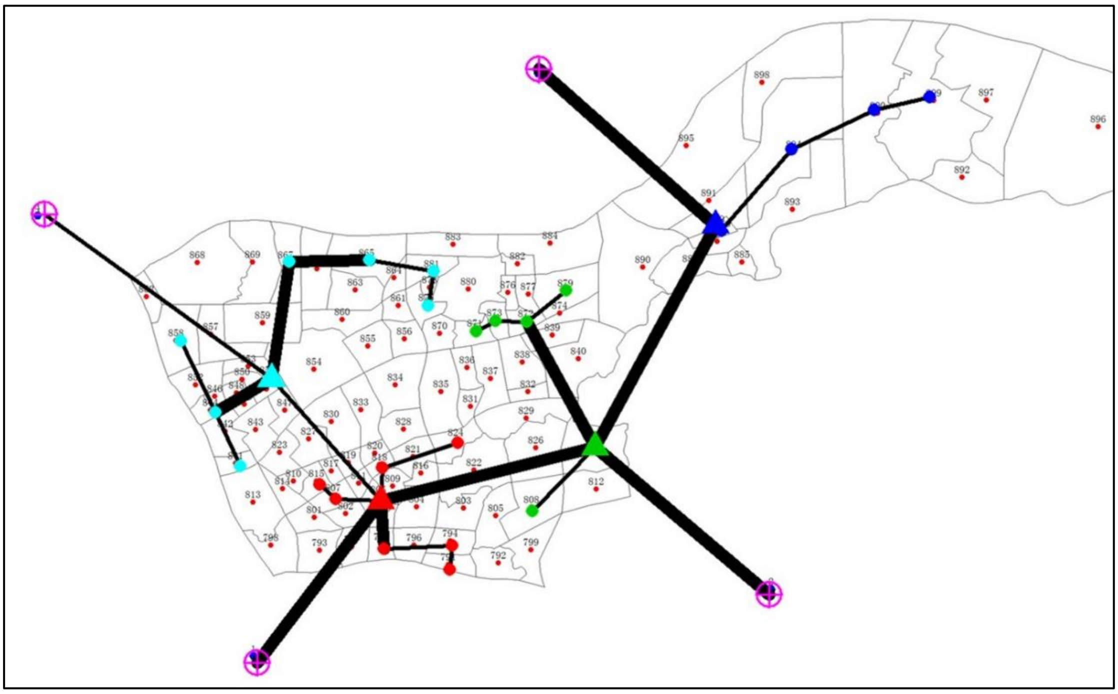

图 6.1.1 成本最低方案示意图

该最优方案的日均成本组成如下表所示：

表 6.1.2 方案日均成本说明

<table>
<tr><th colspan="4">最优情况造价费用名目</th><th>具体项目</th><th>日均成本（万元）</th></tr>
<tr><td rowspan="7">总费用合计</td><td rowspan="7">228.6 万元/日</td><td rowspan="4">建造成本</td><td rowspan="4">117.55 万元/日</td><td>节点</td><td>7.95</td></tr>
<tr><td>园区至一级节点隧道</td><td>43.71</td></tr>
<tr><td>一级节点间隧道</td><td>21.84</td></tr>
<tr><td>大型区域内通道</td><td>44.05</td></tr>
<tr><td rowspan="3">运输成本</td><td rowspan="3">111.05 万元/日</td><td>园区到一级节点流量</td><td>59.55</td></tr>
<tr><td>大型区域内流量</td><td>30.99</td></tr>
<tr><td>一级节点间流量</td><td>20.51</td></tr>
</table>

与问题二得到的设计方案相比，该方案除节点造价与问题二中方案相同外，其余各项成本均有所下降。日建造成本下降 11.23 万元，日运输成本下降 8.30 万元，日总成本下降 19.53 万元。

6.2 地下物流系统面对货流量激增的抗风险性能

当因为突发原因，某线路上的货流量突然增大时，可以通过其它路径分担，这要求地下物流网中尽可能多形成环路。但即便具有多条可以通行的路径，各个路径盈余的货运能力也是十分有限的。要实质性地提高地下物流系统面对货流量激增时的抗风险性能，就应当适度建造更多的节点和隧道，从而提高整个物流网络设计方案的货运量阈值。

地下物流系统的货运量阈值主要受到两个量的限制：节点从地面收发货物的上限和双向四轨隧道的单向日流量上限。为了提升系统的货运量阈值，可以在设计系统时对这两个限制条件按照更为严苛的标准来考虑。具体标准可根据需要提高系统抗风险性能的比率来确定。

在此题中我们拟将整个系统的货运量阈值提升 20%。为此我们按照在当前状况下每个节点从地面收发量不超过 2500 吨/日、线路单向流量最大不超过 6000 吨/日的标准来设计系统。除了这两个限制条件的取值以外，其余部分按照 6.1 节的方法进行，从符合要求的方案中选取出总成本最低的方案。

在当前设定下，共需要 31 个节点。其中一级节点 4 个，分别位于区域 795、808、879、858 中心处。二级节点 27 个，它们与一级节点的对应关系如下表所示：

表 6.2.1 一二级节点从属关系

一级节点	下辖二级节点
795	794、795、806、807、810、818、834
808	799、808、824、829、831
879	837、870、873、875、877、879、886、888、889、894、899、900
858	841、844、849、854、858、860、866

这一方案里，园区至一级节点之间四条线路中有三条使用双向四轨隧道：园区 1—一级节点 1(795)、园区 2—一级节点 2(808)、园区 3—一级节点 3(879)；园区 4—一级节点 4(858)使用双向双轨隧道。

一级节点间的通道（795-808、795-858、808-879）均采用双向四轨隧道。

该方案的地下物流系统示意图如下：

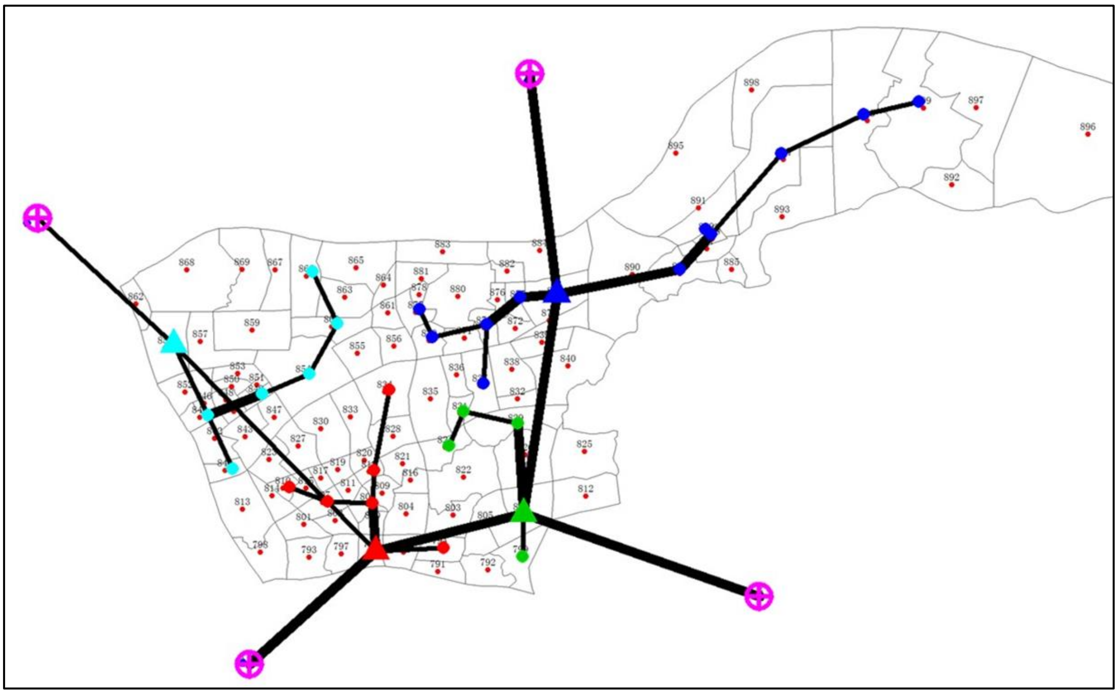

图 6.2.1 抗增流风险的规划方案示意图

经过多次尝试，我们发现当将限制条件调整为节点货运量不超过 2500 吨、节点间线路流量不超过 6000 吨时，即阈值提升 20%时，不仅可以提高抗风险能力，总成本也会出现较大幅度降低。该地下物流系统方案的日均成本组成如下表

所示：

表 6.2.2 方案日均成本说明

最优情况造价费用名目				具体项目	日均成本（万元）
总费用合计	211.48 万元/日	建造成本	116.73 万元/日	节点	9.04
				园区至一级节点隧道	38.89
				一级节点间隧道	21.44
				大型区域内通道	47.36
		运输成本	94.75 万元/日	园区到一级节点流量	54.51
				大型区域内流量	22.21
				一级节点间流量	18.03

6.3 地下物流系统面对线路中断的抗风险性能

当由于不可抗拒的因素发生线路中断时，我们可以将地下货流转至地上运输或将中断线路的货物流分配到其他通行线路运送。由于地面上的运输能力相对固定，如果有大量货流从地下转至地上，必然会导致路面交通拥堵。因此，具有一定抗风险性能的地下物流系统在不相邻的两个节点间应有尽可能多的路径。

从大型区域间线路的层面上看，当前方案的三条一级节点间通道是连通四个大型区域的所需的最少通道条数，若这三条通道中某一条中断，尤其位于中间的一条通道中断，其影响范围将十分大。如下图中(1)图方案，如果 1-2 线路中断，则 1-2、1-3、2-4、3-4 之间的货物运输都会被阻滞，将有大量货物需要地面运输解决。如下图中(2)图的方案，任何两个区域之间都有两种运输路径，任何一条隧道中断，货物还可以通过另一条路径进行地下运输，对地面交通造成的压力较小。这种连接方案抗风险性能高，但缺点是其成本很高。如下图中图(3)的连接方案，相比于图(1)而言，通道的总长度有所增加，但增量不多；虽然每两点之间仍然只有一条路径，但在任意一条线路中断的情况下，受到影响的范围相对较小。

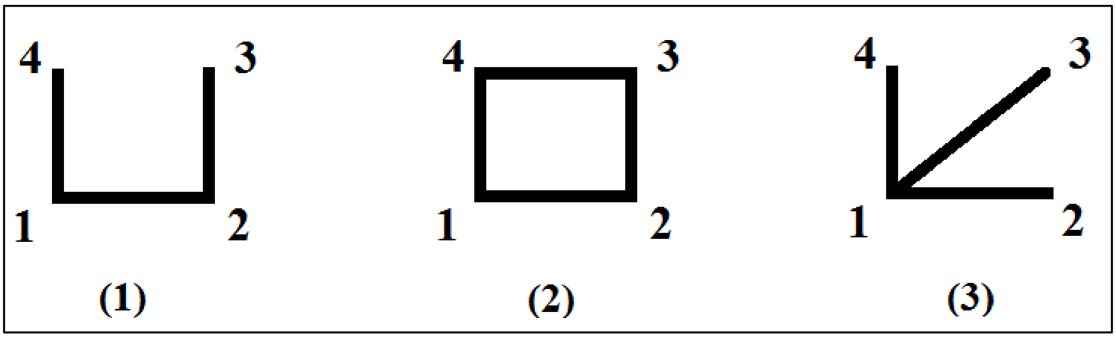

图 6.3.1 大型区域间通道的抗中断风险性能

从园区到一级节点间线路的层面上看，由于园区距离仙林区市区的距离较远，增设园区到一级节点的线路成本最高，且需要同时增加 4 条线路才能较好处理此

类型线路中断造成的问题，成本增加过大，性价比过小，暂时不考虑改进。

从大型区域内部的线路层面上看，应当在地下物流系统网络中尽可能多地形成环路，当环路中的线路发生中断时，货物始终可以通过适当绕行进行运送。因此，如需提升地下物流系统面对线路中断的抗风险能力，可以在上文设计出的地下物流网基础上在大型区域内一些距离较近但不直接相连的节点间增修线路，形成环路。

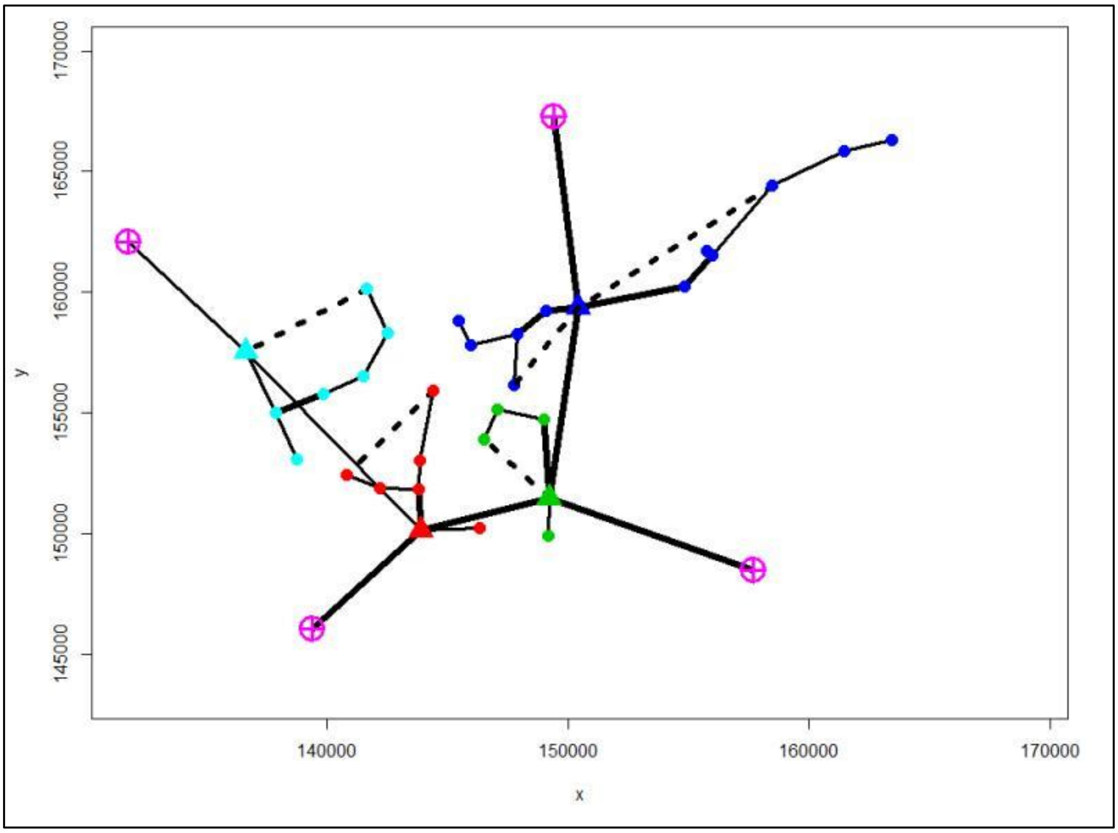

图 6.3.2 抗中断风险的规划方案示意图

上图给出一种增修方案。由虚线表示的是我们新增的 5 条线路。新增的线路与原有线路共同形成了多个环形线路。由于环形线路上任何两点之间都存在两种连通路径，在环线上有一条隧道中断时，可基本保证各点之间的运输不受影响。

七、问题 4 的解答

7.1 目标和假设

针对地下物流系统造价高，风险大，改建困难的特点，我们应从动态的眼光考虑多方面因素，而不能简单的认为现有的状态会保持静止不变。问题 4 要求建造的地下物流系统能满足该市近 30 年内的交通需求，且假设该区域交通需求量以每年 5%的速度增长，因此我们规划的地下物流系统应该不仅能满足现在的交通状况，还要求我们从动态规划的角度，使得其在未来也不会失效。此外，由于建造工期可能会长达八年，在这八年里交通状况也是会发生改变的，所有优先建造哪些节点、隧道也是值得研究的问题，文献[9]，[10]对此问题有较深入的研究。

针对本问题，我们首先预估 30 年后各区域的交通拥堵情况和需要缓解的交通流量，假设地面运输能力由于新建道路等因素也以每年 5%的速度增长，则各区域需要缓解的交通流量为问题 1 中（表 4.1.1）额外吞吐量的 1.05^30=4.32 倍。在如此规模的增速下，如果仍假设各节点吞吐能力为 3000 吨（二级）、4000 吨（一级），则即使所有 110 个区域都有一个节点，也无法满足需要缓解的交通流量，即保持原假设会导致没有可行解。因此考虑到技术也在不断增长，我们假设二级节点吞吐能力增长一倍为 6000 吨，一级节点同样增长为 8000 吨，同时由于单条隧道运送能力的增长可以忽略不计，关于隧道及车辆的运输能力仍维持原假设。

7.2 动态的节点和线路的规划

考虑到节点数目大量增加，如果仍只用四个一级节点，将会造成以一级节点为核心形成的大型区域过于庞大的弊端，因此，除了服务四个园区的一级节点（记为服务型一级节点），我们新增了一个起中转作用的一级节点（记为中转型一级节点）。除此之外，在对各大区内部进行隧道规划时，如果仍采用最小生成树法，由于每个大区域内的地下流量额度的增加，难以避免的会出现隧道流量超标的情况。因此，对大区内部我们采用寻找欧拉回路的方法，这样由于构成环路，也增强了抗风险能力。

具体操作而言，首先仍用第一题的混合整数规划，得到节点的分布，分布图如下，共计 55 个节点（一二级节点及所辖范围是之后确定的）。

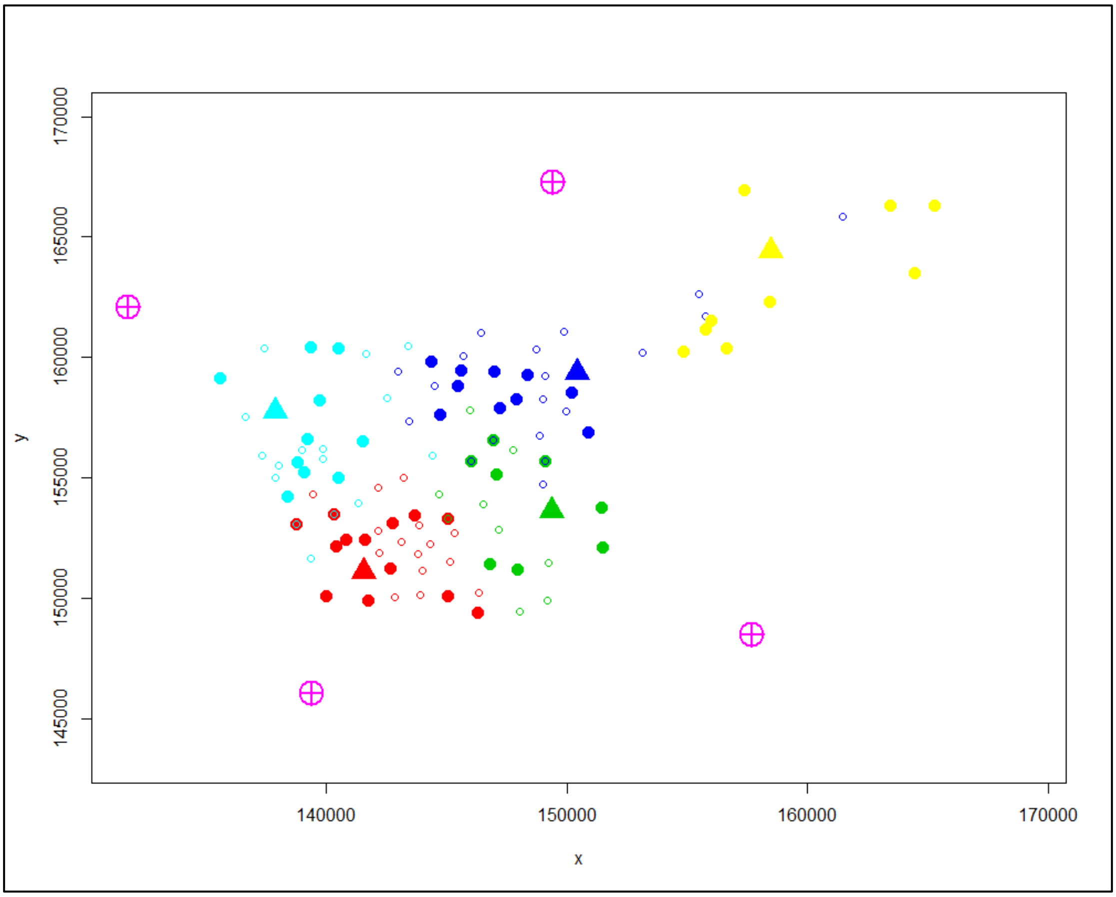

图 7.2.1 节点分布示意图

之后我们采用问题三中的全局分步规划算法：首先同样将离园区最近的五个

点设为该园区对应服务型一级节点的候选点，然后遍历这 5^4=625 种情况。

对每种情况组合，刨去四个一级节点所属区域外，还余下 106 个区域，我们知道这 106 个区域中心点的坐标和每个区域的流量，则可以把这 106 个中心点看成是 xy 平面上的分布点，采用**<u>核密度估计</u>**的方法估计出整个所有区域内每一点的流量分布的核密度，我们采用如下标准选取余下的中转型一级节点：从余下的 51 个二级节点中选出流量核密度最大的五个二级节点（流量核密度越大，代表该处流量越大，越需要提高额度，修建一级节点）作为备选，再从五个备选节点中选出和四个服务型一级节点距离平方和最大的一个节点作为中转型一级节点，这样做是为了保证一级节点间距离不会太近。

对每种情况组合，采用如上方法选定中转型一级节点后，仍采用就近原则分配二级节点，形成五个大区（每大区一个一级节点），对大区内部，采用 fleury 算法求出该大区的最短欧拉回路，将回路对应的二级节点相连，同时视每条隧道的流量决定其规模（双轨/四轨）。

而一级节点间的隧道选择方面，则同样用 fleury 算法找出五个一级节点的最大回路并相连，视流量是否超标决定是否添加其他线路。

当节点和隧道给定后，采用和第二题同样的方法计算总成本，如此遍历 625 种情况，找出其中日均总成本最小的作为我们的方案，最优方案示意图如下，蓝色为增加的中转型一级节点。

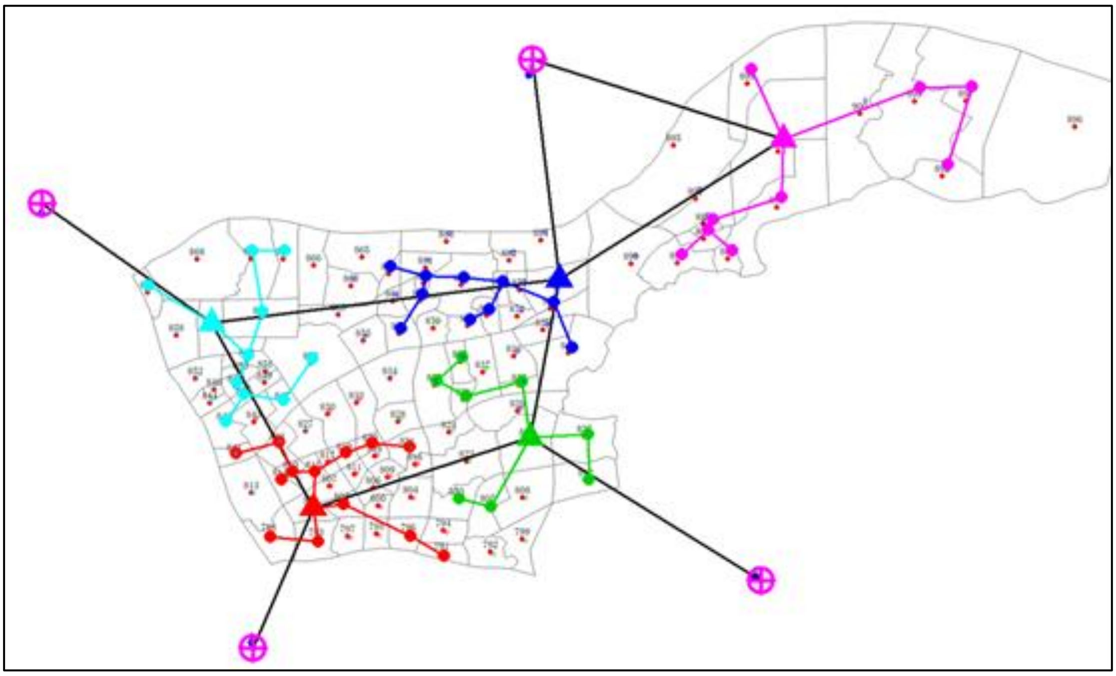

图 7.2.2 地下物流系统网络示意图

7.3 时序与动态优化的建设原则

由于每年只能建设隧道总长度的八分之一，所以我们在确定方案后仍需考虑优先建造哪些节点和隧道，即建设时序。我们引入**<u>节点重要度</u>**的概念并以此为标准选择建造顺序。

重要度判断规则如下（按先后顺序）：

1)和已有任一节点或园区相连接的隧道在八年规划建设的隧道中的节点和任意节点或园区相连接的隧道都不在八年规划建设的隧道中的节点；

2)一级节点>二级节点；

3)对于同种节点（一级和一级，二级和二级），八年规划建设完毕后，通过此节点预估的流量越大的节点，重要性越高。按照上述三条标准得到所有节点的重要度排序，则每年建造的节点为所有未建造节点中重要度从高往低排的 7 个节点（55/8）及其对应隧道。

相比第三题设置的网络，其节点层次（重要度）更加分明，抗风险能力进一步提升（由于大区域内部采用寻找欧拉回路法而非最小生成树法）。并且随着需求量逐年增长，我们第三题中设计的网络虽然增加了 20%的流量阈值，但是仍会在第四年时流量达到饱和，故从长远的动态规划的角度，建议采用第四题的方案。

参考文献

[1] 陈端兵,黄文奇,一种求解集合覆盖问题的启发式算法,计算机科学,(04):133-136,2007.

[2] 穆树录,城市地下物流系统线路规划研究,石家庄铁道大学,1-68，2015.

[3] 彭玫贞,陈一村，城市地下物流系统探析，江苏科技信息，(19):65-67，2017.

[4] 权光日,洪炳熔,叶风,任世军,集合覆盖问题的启发函数算法,软件学报,(02):77-81, 1998.

[5] 孙焰，现代物流管理技术——建模理论及算法设计，上海：同济大学出版社，1-231，2004.

[6] 徐国峰,城市地下物流系统架构研究，华中科技大学，1-126，2012.

[7] 闫文涛,城市地下物流系统节点选址研究,重庆交通大学,1-51,2015.

[8] 章冠弘，包敦辉，城市地下物流系统研究概况与分析，价值工程，36(29):242-245，2017

[9] 马成林. 我国大城市地下物流系统规划关键技术研究[D]. 东南大学, 2011.

[10] 王立亮. 基于复杂网络的城市道路建设时序优化研究[D]. 重庆交通大学, 2014.

[11] 姜阳光, 庞大钧. 基于集合覆盖模型的城市 ULS 物流节点选址分析[J]. 物流科技, 2009, 32(10): 54-55.

附件 Matlab 及 R 代码

1. Matlab 代码

1-1 问题一

```
clc;
clear;
%%
Distance=xlsread('data','Distance','A1:DF110');
adjusted=xlsread('data','adjusted','A1:A110');
adjusted=adjusted.*1.05^30;
N=6000;%2500
leng=length(adjusted);
%%
f = [ones(leng,1);zeros(leng*leng,1)];
intcon = [1:leng]';
A=zeros(2*leng,leng*(leng+1));
A(1:leng,1:leng)=zeros(leng,leng);
for i=1:leng
    A(1:leng,(leng*i+1):(i+1)*leng)=-diag(Distance(:,i));
end
A(leng+1:leng*2,1:leng)=-N*eye(leng);
for i=1:leng
    A(leng+i,(leng*i+1):(i+1)*leng)=(Distance(:,i).*adjusted)';
end
b=[-1*ones(leng,1);zeros(leng,1)];%Ax<=b
% b(76)=0;b(105)=0;b(106)=0;
Aeq=[];
beq=[];
Ib=zeros(leng*(leng+1),1);
ub=ones(leng*(leng+1),1);
[phi,fval,exitflag,output] = intlinprog(f,intcon,A,b,Aeq,beq,Ib,ub);
x_result=phi(1:leng);
number=sum(x_result);
y_result=phi(leng+1:end);
matrix_y=reshape(y_result,[110,110]);
```

1-2 问题二

```
clc;
clear;
%% 3000 图
a1=xlsread('data','quyujuli','B2:H8');
b1=xlsread('data','quyujuli','B9:H9');
```

```
result1=MST_Prim(a1);%3
a2=xlsread('data','quyujuli','K2:O6');
b2=xlsread('data','quyujuli','K7:O7');
result2=MST_Prim(a2);%1
a3=xlsread('data','quyujuli','B12:H18');
b3=xlsread('data','quyujuli','B19:H19');
result3=MST_Prim(a3);%4
a4=xlsread('data','quyujuli','K12:r19');
b4=xlsread('data','quyujuli','k20:r20');
result4=MST_Prim(a4);%4
% %% 3000 图
% UG = sparse(a2);
% [ST,pred] = graphminspantree(UG,4);
% view(biograph(ST,[],'ShowArrows','off','ShowWeights','on'));
% %% 2500 图
% a1=xlsread('data','quyujuli2500','B2:I9');
% b1=xlsread('data','quyujuli2500','B10:I10');
% result1=MST_Prim(a1);%2
% a2=xlsread('data','quyujuli2500','l2:p6');
% b2=xlsread('data','quyujuli2500','l7:p7');
% result2=MST_Prim(a2);%1
% a3=xlsread('data','quyujuli2500','B13:i20');
% b3=xlsread('data','quyujuli2500','B21:i21');
% result3=MST_Prim(a3);%5
% a4=xlsread('data','quyujuli2500','l13:u22');
% b4=xlsread('data','quyujuli2500','l23:u23');
% result4=MST_Prim(a4);%7
%%  图 1
result1(4,1)=b1(1);
result1(4,2)=b1(5);
result1(4,3)=b1(5)+b1(1)+b1(2);
result1(4,4)=b1(4)+b1(6)+b1(7);
result1(4,5)=b1(6)+b1(7);
result1(4,6)=b1(7);
%%  图 2
result2(4,1)=b2(2)+b2(3)+b2(4)+b2(5);
result2(4,2)=b2(3)+b2(4)+b2(5);
result2(4,3)=b2(5);
result2(4,4)=b2(4);
%%  图 3
result3(4,1)=b3(1);
result3(4,2)=b3(1)+b3(2);
```

```
result3(4,3)=b3(3)+b3(5)+b3(6)+b3(7);
result3(4,4)=b3(5)+b3(6)+b3(7);
result3(4,5)=b3(6)+b3(7);
result3(4,6)=b3(6);
%% 图 4
result4(4,1)=b4(2);
result4(4,2)=b4(1)+b4(2);
result4(4,3)=b4(3)+b4(5)+b4(6)+b4(7)+b4(8);
result4(4,4)=b4(5)+b4(6)+b4(7)+b4(8);
result4(4,5)=b4(5)+b4(6)+b4(7);
result4(4,6)=b4(5)+b4(6);
result4(4,7)=b4(6);
% 在 3600-7200 之间用四车道的线：result1-(2,3),result1-(3,4)；result2-(1,2),result2-(2,3)；result3-(3,4);result4-(3,4);result4-(3,8);result4-(7,8);
%% 计算价格,单位万
%节点造价
cost1=(1.5*4+1*23)*0.01*1/365*10000;
%园区到一级节点隧道造价
cost2=(5*(6.49+11.45+8.43)+4*8.92)*0.01*1/365*10000;
%一级节点和二级节点间隧道费用
cost3=0.5*(3*(sum(result1(3,:))+sum(result2(3,:))+sum(result3(3,:))+sum(result4(3,:)))+0.5*(result1(3,3)+result1(3,4)+result2(3,1)+result2(3,2)+result3(3,3) ...
    +result4(3,3)+result4(3,4)+result4(3,5)))*0.01*1/365*10000;
%一级节点间隧道流量：两部分，一部分是园区和三大区域的交互量*转运率，另一部分四大区域所有非园区部分的交互量*地下估计率，...
%这俩比率不同是因为第一个问尽量让进园区内部消化的走地下,转运率会低一点，所以四大区域间走地上比率高（园区部分）；估计这个比率的方法是：
%max(每个区域地下额度-该区域与 4 园区交互值,0)为其非园区地下额度值，加和再除以四大区域所有非园区交互值总和，为其地下估计率
%加和值为(76194.57847-72307.06097)=3888，3888/69082=5.63%
%经比较，由于隧道建造费用远大于流量费（10 万+)vs（不超过 2 万），所以以道路最短为方案，4-1-2-3;1-4:8.66+2-3:15.5+1-2:4.8
cost4=3.5*(8.66+15.5+4.8)*0.01*1/365*10000;
%园区到一级节点地下运输费用
cost5=(21176*6.49+20845*11.45+20535*8.43+9751*8.92)/10000;%四个园区地下吞吐量，见 word
%一级节点和二级节点间地下运输费用
cost6=(sum(result1(3,:).*result1(4,:))+sum(result2(3,:).*result2(4,:))+sum(result3(3,:).*result3(4,:))+sum(result4(3,:).*result4(4,:)))/10000;
%一级节点间运输费用，4-1-2-3
```

```
cost7=(7896*8.7+10089*4.8+7208*15.5)/10000;

cost=cost1+cost2+cost3+cost4+cost5+cost6+cost7;
cost_fixed=cost1+cost2+cost3+cost4;
```

1-3 问题二计算最小生成树函数

```
function result=MST_Prim(a)
%问题：最小生成树(Minimum Spanning Tree)
%算法：Prim

% clc, clear;
%% 做连接图
% a=xlsread('data','quyu','B2:H8');
% a2=xlsread('data','quyu','K2:O6');
% a3=xlsread('data','quyu','B12:H18');
% a4=xlsread('data','quyu','K12:r19');
%% Prim 算法 （思考）
a=a+a';a(a==0)=inf;
result=[];p=1;tb=2:length(a);
while length(result)~=length(a)-1
    temp=a(p,tb);temp=temp(:);
    d=min(temp);
    [jb,kb]=find(a(p,tb)==d);
    j=p(jb(1));k=tb(kb(1));
    result=[result,[j;k;d]];p=[p,k];tb(tb==k)=[];
end
Wt=sum(result(3,:))/2;
disp(['最短架设电线总长度：', int2str(Wt)]);
%% 画最小树
axis equal;                    %画最小生成树
hold on
n=length(a);
[x,y]=cylinder(1,n);      %画出顶点，均匀画圆
xm=min(x(1,:));
ym=min(y(1,:));
xx=max(x(1,:));
yy=max(y(1,:));
axis([xm-abs(xm)*0.15,xx+abs(xx)*0.15,ym-
abs(ym)*0.15,yy+abs(yy)*0.15]);
plot(x(1,:),y(1,:),'ko');
for i=1:n
    temp=['v',int2str(i)];
```

```
        text(x(1,i),y(1,i),temp);
end
% for i=1:k-n+1 %画出不在树内的边
%           plot(x(1,data(1:2,i)),y(1,data(1:2,i)),'b');
% end
for i=1:n-1    %画出树内的边
        plot(x(1,result(1:2,i)),y(1,result(1:2,i)),'r');
end
text(-0.35,-1.2,['最小生成树的权为','',num2str(Wt)]);
title('红色连线为最小生成树');
axis off;
hold off;
end
```

（4）计算距离

```
clc;clear;
xy=xlsread('data','xyindex2500','B2:C115');
[m,n]=size(xy);
for i=1:m
        for j=1:m
                A(i,j)=sqrt((xy(i,1)-xy(j,1))^2+(xy(i,2)-xy(j,2))^2);
        end
end
D=786;
a1=A([794-D,795-D,806-D,807-D,810-D,818-D,824-D,841-D],[794-D,795-D,806-D,807-D,818-D,824-D,810-D,841-D]);
a2=A([799-D,808-D,829-D,831-D,837-D],[799-D,808-D,829-D,831-D,837-D]);
a3=A([877-D,879-D,886-D,888-D,889-D,894-D,899-D,900-D],[877-D,879-D,886-D,888-D,889-D,894-D,899-D,900-D]);
a4=A([834-D,844-D,849-D,854-D,858-D,860-D,866-D,870-D,873-D,875-D],[834-D,844-D,849-D,854-D,858-D,860-D,866-D,870-D,873-D,875-D]);
```

2. R 代码

```
library(readxl)
library(plotrix)
library(igraph)
library(readxl)
data1=read.csv("C:/Users/mike/Desktop/gmcm2017/2017 年试题/2017 最终赛题/F/2017 年中国研究生数学建模竞赛 F 题/工作簿 1.csv",header=T)
class(data1)
data1=as.matrix(data1[,-1])
```

```
data[1:5,1:5]
dim(data1)
ll=rowSums(data)+colSums(data)
sum(data[1,]);sum(data[,1])
ll[1:4]

need=read.csv("C:/Users/mike/Desktop/need.csv")[,2]
data2=read.csv("C:/Users/mike/Desktop/gmcm2017/2017 年试题/2017 最终赛题/F/2017 年中国研究生数学建模竞赛 F 题/现状 OD 数据及其他数据.csv",header=T)
data22=data2[-c(1:4),]
dim(data2)-4
A=matrix(,110,110)
for (i in 1:110) {
  for (j in 1:110) {
    A[i,j]=as.numeric(sum((data2[i+4,4]-data2[j+4,4])^2+
                           (data2[i+4,5]-data2[j+4,5])^2)<=9000000)
  }
}
# 898 894


write.csv(A,"C:/Users/mike/Desktop/A.csv")
write.csv(need,"C:/Users/mike/Desktop/need.csv")
id=read.csv("C:/Users/mike/Desktop/id.csv",header = F)
rowSums(A)



need=NA
for (i in 5:114) {
  if(tran[i]<=4)
    need[i-4]=0
  else
    need[i-4]=ll[i]-ll[i]/tran[i]*4
}
tran[tran<=4]



y=as.matrix(read_excel("C:/Users/mike/Desktop/data(1).xlsx",sheet =
4,col_names=F))
```

```
dim(y)


x=read_excel("C:/Users/mike/Desktop/data(1).xlsx",sheet =
3,col_names=F)[,1]


#x=read_excel("C:/Users/mike/Desktop/6000data.xlsx",sheet =
1,col_names=F)[,1]
class(x)
dim(x)
sum(x)
index=matrix(,4,sum(x))
ro=3
for(i in 1:4)
{
  index[i,]=which(x==1)[order((data2[i,4]-data22[x==1,4])^2+(data2[i,5]-
data22[x==1,5])^2)[1:ro]]
}

un.ratio=72307.06097/128865.538
t=1
data22=data2[-c(1:4),]
sum.trans.ratio=NA
clas=matrix(NA,ro^4,sum(x))
for (i in 1:ro) {
  for (j in 1:ro) {
    for (k in 1:ro) {
      for (l in 1:ro) {
        for (m in 1:sum(x)) {
          clas[t,m]=which.min((data22[x==1,4][m]-
data22[c(index[1,i],index[2,j],index[3,k],index[4,l]),4])^2+
                  (data22[x==1,5][m]-
data22[c(index[1,i],index[2,j],index[3,k],index[4,l]),5])^2)
            }

sum.trans.ratio[t]=min(sum((need%*%y)[x==1][clas[t,]==1])/2,sum((data1
[1,-c(1:4)]%*%y)[x==1][clas[t,]==1]))/sum(data1[1,])*

min(sum((need%*%y)[x==1][clas[t,]==2])/2,sum((data1[2,-
c(1:4)]%*%y)[x==1][clas[t,]==2]))/sum(data1[2,])*
```

```
min(sum((need%*%y)[x==1][clas[t,]==3])/2,sum((data1[3,-
c(1:4)]%*%y)[x==1][clas[t,]==3]))/sum(data1[3,])*

min(sum((need%*%y)[x==1][clas[t,]==4])/2,sum((data1[4,-
c(1:4)]%*%y)[x==1][clas[t,]==4]))/sum(data1[4,]);
        t=t+1

      }
    }
  }
}
t=which.max(sum.trans.ratio)
#t=11
o=t
o

trans.ratio=un.ratio-
c(min(sum((need%*%y)[x==1][clas[t,]==1])/2,sum((data1[1,-
c(1:4)]%*%y)[x==1][clas[t,]==1]))/sum(data1[1,])
              ,min(sum((need%*%y)[x==1][clas[t,]==2])/2,sum((data1[
2,-c(1:4)]%*%y)[x==1][clas[t,]==2]))/sum(data1[2,])
              ,min(sum((need%*%y)[x==1][clas[t,]==3])/2,sum((data1[
3,-c(1:4)]%*%y)[x==1][clas[t,]==3]))/sum(data1[3,])
              ,min(sum((need%*%y)[x==1][clas[t,]==4])/2,sum((data1[
4,-c(1:4)]%*%y)[x==1][clas[t,]==4]))/sum(data1[4,]))
trans.ratio
sum(trans.ratio)
oo=o-1
ic=c(oo%/%ro%/%ro%/%ro%%ro,oo%/%ro%/%ro%%ro,oo%/%ro%%r
o,oo%%ro)+1
ic
first=c(index[1,ic[1]],index[2,ic[2]],index[3,ic[3]],index[4,ic[4]])

first+790
(need%*%y)[x==1][clas[o,]==1]
(need%*%y)[x==1][clas[o,]==2]
(need%*%y)[x==1][clas[o,]==3]
(need%*%y)[x==1][clas[o,]==4]

which(x==1)[clas[o,]==1]+790
which(x==1)[clas[o,]==2]+790
```

```
which(x==1)[clas[o,]==3]+790
which(x==1)[clas[o,]==4]+790
#plot
plot(data2[,4],data2[,5],asp=1,xlab = "x",ylab = "y")
points(data2[1:4,4],data2[1:4,5],pch=10,col=6,cex=3,lwd=2)
points(data2[which(need==0)+4,4],data2[which(need==0)+4,5],col="blue",
pch=13)
for(i in 1:110)
{ if(x[i]>0)
   {
     points(data2[i+4,4],data2[i+4,5],col="red",pch=16,cex=1.5)
     draw.circle(data2[i+4,4],data2[i+4,5],3000,lty =2)
      }
}




#first ##一级节点

plot(data2[,4],data2[,5],type="n",asp=1,xlab = "x",ylab = "y")
points(data2[1:4,4],data2[1:4,5],pch=10,col=6,cex=3,lwd=2)
data22[rowSums(y[,x==1][,clas[o,]==i])>0,4]
for(i in 1:4)
{ #if(any(y[,y[,x==1]])

points(data22[rowSums(y[,x==1][,clas[o,]==i])>0,4],data22[rowSums(y[,x=
=1][,clas[o,]==i])>0,5],col=i+1)
   points(data22[first[i],4],data22[first[i],5],col=i+1,pch=17,cex=2.5)

points(data22[x==1,][clas[o,]==i,4],data22[x==1,][clas[o,]==i,5],col=i+1,pch
=16,cex=1.5)
}
cl5=data22[x==1,1]>884
points(data22[x==1,][cl5,4],data22[x==1,][cl5,5],col=7,pch=16,cex=1.5)
points(data22[894-790,4:5],col=7,pch=17,cex=2.5)


##########轨道图#################
####################################
```

```
for(i in 1:4)
{ #if(any(y[,y[,x==1]])

#points(data22[rowSums(y[,x==1][,clas[o,]==i])>0,4],data22[rowSums(y[,x
==1][,clas[o,]==i])>0,5],col=i+1)
   points(data22[first[i],4],data22[first[i],5],col=i+1,pch=17,cex=2.5)

points(data22[x==1,][clas[o,]==i,4],data22[x==1,][clas[o,]==i,5],col=i+1,pch
=16,cex=1.5)
}

########################################
#######################################


##求四大区交互流量
############################################
+0.56*sum(data1[i,cc[j,]]
cc=matrix(,4,110)
               #cc=NULL
for (ni in 1:4)
{ cc[ni,]=rowSums(y[,x==1][,clas[t,]==ni])>0
               }
flow=matrix(,4,4)
for(i in 1:4)
{
   for(j in 1:4)
   { flow[i,j]=(sum(data1[-c(1:4),-c(1:4)][cc[i,],cc[j,]])+
                          sum(data1[-c(1:4),j][cc[j,]]))*0.1586
          trans.ratio[i]*sum(data1[i,-c(1:4)][cc[j,]])
   }
}
diag(flow)=0
flow

flow=matrix(,4,4)
for(i in 1:4)
{
   for(j in 1:4)
   { flow[i,j]=trans.ratio[i]*sum(data1[i,-c(1:4)][cc[j,]])
   }
}
```

flow

```
##########################################


cc=matrix(,4,110)
for (i in 1:4) {
  cc[i,]=rowSums(y[,x==1][,clas[o,]==i])>0
}

##class(get.edgelist(sm))



####################################################
########
####################################################
########
###################轨道图
c1=list()
for(i in 1:4)
{c1[[i]]=which(x>0)[clas[o,]==i]
}
plot(data2[,4],data2[,5],type="n",asp=1,xlab = "x",ylab = "y")
for(i in 1:4)
{
fir=data22[x==1,][clas[o,]==i,4:5]
di<-as.matrix(dist(as.matrix(fir),diag = F,upper=T))
h=make_full_graph(dim(di)[1])
graph_attr(h,'weight')=di[lower.tri(di)==1]
sm<-mst(h,weights =graph_attr(h,'weight'))
edg=get.edgelist(sm)
which(first[1]==c1[[1]])
lines(data2[c(i,c1[[i]][which(first[i]==c1[[i]])]+4),4],
data2[c(i,c1[[i]][which(first[i]==c1[[i]])]+4),5],col=15,lwd=8)
for(j in 1:length(E(sm)))
 {
  lines(data22[c1[[i]][edg[j,]],4], data22[c1[[i]][edg[j,]],5],col=15,lwd=4)
 }
}
points(data2[1:4,4],data2[1:4,5],pch=10,col=6,cex=3,lwd=2)
for(i in 1:4)
```

```
{ #if(any(y[,y[,x==1]])

#points(data22[rowSums(y[,x==1][,clas[o,]==i])>0,4],data22[rowSums(y[,x
==1][,clas[o,]==i])>0,5],col=i+1)
   points(data22[first[i],4],data22[first[i],5],col=i+1,pch=17,cex=2.5)

points(data22[x==1,][clas[o,]==i,4],data22[x==1,][clas[o,]==i,5],col=i+1,pch
=16,cex=1.5)
}
#####################################################
########
#####################################################
########
c1=list()
for(i in 1:4)
{c1[[i]]=which(x>0)[clas[o,]==i]
}
plot(data2[,4],data2[,5],type="n",asp=1,xlab = "x",ylab = "y")
for(i in 1:4)
{
   fir=data22[x==1,][clas[o,]==i,4:5]
   di<-as.matrix(dist(as.matrix(fir),diag = F,upper=T))
   h=make_full_graph(dim(di)[1])
   graph_attr(h,'weight')=di[lower.tri(di)==1]
   E(h)$weight=di[lower.tri(di)==1]
   sm<-mst(h,weights =graph_attr(h,'weight'))
   plot(sm)
   sm<-mst(h)
   plot(sm)

   plot(h)
   make_full_graph(,weight= )
   ?make_full_graph
   edg=get.edgelist(sm)
   which(first[1]==c1[[1]])
?make_graph
plot(sm)
   class(shortest_paths(sm,1))
  class( shortest_paths(sm,1)[[1]][[3]])
  distances(sm,3)
  sm
  class(distances(sm,1))
```

```
####################################################3
3
 ro=3
 for(i in 1:4)
 {
    index[i,]=which(x==1)[order((data2[i,4]-data22[x==1,4])^2+(data2[i,5]-
data22[x==1,5])^2)[1:ro]]
 }
 un.ratio=72307.06097/128865.538


ro=5
t=513
oo=t-1
ic=c(oo%/%ro%/%ro%/%ro%%ro,oo%/%ro%/%ro%%ro,oo%/%ro%%r
o,oo%%ro)+1
i=ic[1];j=ic[2];k=ic[3];l=ic[4]
i=ic[1];j=ic[2];k=ic[3];l=ic[4]
clas=matrix(NA,ro^4,sum(x))
cost=matrix(,ro^4,5)
index=matrix(,4,ro)
i=1
for(i in 1:4)
{
   index[i,]=which(x==1)[order((data2[i,4]-data22[x==1,4])^2+(data2[i,5]-
data22[x==1,5])^2)[1:ro]]
}
t=1
#max(table(clas[1,]))-min(table(clas[1,]))

for (i in 1:ro) {
   for (j in 1:ro) {
      for (k in 1:ro) {
         for (l in 1:ro) {
            for (m in 1:sum(x) {
               clas[t,m]=which.min((data22[x==1,4][m]-
data22[c(index[1,i],index[2,j],index[3,k],index[4,l]),4])^2+
                                    (data22[x==1,5][m]-
data22[c(index[1,i],index[2,j],index[3,k],index[4,l]),5])^2)
            }
```

```
            trans.ratio=un.ratio-
c(min(sum((need%*%y)[x==1][clas[t,]==1])/2,sum((data1[1,-
c(1:4)]%*%y)[x==1][clas[t,]==1]))/sum(data1[1,])
                                        ,min(sum((need%*%y)[x==1][clas[t,]=
=2])/2,sum((data1[2,-c(1:4)]%*%y)[x==1][clas[t,]==2]))/sum(data1[2,])
                                        ,min(sum((need%*%y)[x==1][clas[t,]=
=3])/2,sum((data1[3,-c(1:4)]%*%y)[x==1][clas[t,]==3]))/sum(data1[3,])
                                        ,min(sum((need%*%y)[x==1][clas[t,]=
=4])/2,sum((data1[4,-c(1:4)]%*%y)[x==1][clas[t,]==4]))/sum(data1[4,]))
            trans.ratio[trans.ratio<0]=0
            ii=c(i,j,k,l)
            for(nn in 1:4)
            {
               fir=as.matrix(data22[x==1,][clas[t,]==nn,4:5])
               di.2<-as.matrix(dist(fir),diag = F,upper=T)
               h=make_full_graph(dim(di.2)[1])
               E(h)$weight=di.2[lower.tri(di.2)==1]
               V(h)$name<-which(x>0)[clas[t,]==nn]
               ms<-mst(h)
               d=length(V(ms))
               edgelist=list()
               #
               for(ni in 1:d)
               {
                  v=as.character(index[nn,ii[nn]])
                  v.sh=shortest_paths(ms,v,weights = rep(1,(d-
1)))$vpath[[ni]]$name
                 # v.2=v.sh[v.sh-index[nn,ii[nn]]!=0]
                  #if(length(v.2)>0)
                  #{fl=as.numeric(need%*%y)[c(v.2)]}
                  #if(sum(fl)>14400)
                 #    { break}
                 sub.ms=induced_subgraph(ms,vids=as.character(v.sh))
                  edgelist[[ni]]=t(get.edgelist(sub.ms))
               }
               edg=get.edgelist(ms)
               four.edge=matrix(0,d-1,3)
               ntt=1
               f.e=0
               for(ni in 1:nrow(edg))
               {
                  e.n=list()
```

```
      ntj=1
      for(nj in 1:d)
      {
        if(length(edgelist[[nj]])==0){next}
        if(any(apply(edg[ni,]==edgelist[[nj]],2,all)))
        {
          tab= table(as.numeric(edgelist[[nj]]))
          h=as.numeric(names(tab)[tab==1])
          e.n[[ntj]]=t(h[h!=index[nn,ii[nn]]])
          ntj=ntj+1
           }
      }
      if( length(e.n)>0)
        #[!duplicated(ena)]
        ena=as.numeric(as.data.frame(e.n))
      f.e=sum((need%*%y)[ena])
      if(f.e>14400)
      { break
        }
      if(f.e>7200)
      { four.edge[ntt,]=c(edg[ni,],ni)
        ntt=ntt+1
      }
    }
    if(f.e>14400)
    { break}
    #which(x>0)
   el=as.matrix(as.data.frame(edgelist))
    colnames(el)=1:ncol(el)
    n.edge=NULL
    for(np in 1:(d-1))
     {
      n.edge[np]=sum(apply(get.edgelist(ms)[np,]==el,2,all))
    }


dis.1to2=as.numeric(distances(ms,as.character(index[nn,ii[nn]]))/1000)
          dis.1toy=dist(data2[c(nn,index[nn,ii[nn]]+4),4:5])[1]/1000

c21=(sum(E(ms)$weight[four.edge[,3]]*0.5)+sum(E(ms)$weight*3))*1000/
365#####
```

```
        cost[t,nn]=c21+dis.1toy*4000000/365 +
dis.1toy*(sum(data1[nn,])+sum(data1[,nn]))*0.5611+sum(dis.1to2*((need
%*%y)[x==1][clas[t,]==nn]))
        if((sum(data1[nn,])+sum(data1[,nn]))*0.5611>14400)
          cost[t,nn]=cost[t,nn]+dis.1toy*1000000/365
      }
       if(f.e>14400)
       { print(t)
        t=t+1
        next
        }


ff=as.matrix(data22[c(index[1,i],index[2,j],index[3,k],index[4,l]),4:5])
      di.1<-as.matrix(dist(ff),diag = F,upper=T)
      g=make_full_graph(4)
      E(g)$weight=di.1[lower.tri(di.1)==1]
      V(g)$name<-
as.character(c(index[1,i],index[2,j],index[3,k],index[4,l]))
      msg<-mst(g)
      cc=matrix(,4,110)
      for (ni in 1:4) {
        cc[ni,]=rowSums(y[,x==1][,clas[t,]==ni])>0
      }
      flow=matrix(,4,4)
      for(ni in 1:4)
      {
        for(nj in 1:4)
        { flow[ni,nj]=(sum(data1[-c(1:4),-
c(1:4)][cc[ni,],cc[nj,]])+sum(data1[-
c(1:4),][cc[ni,],nj]))*0.1586+trans.ratio[ni]*sum(data1[,-c(1:4)][ni,cc[nj,]])
        }
      }
      diag(flow)=0
      d.flow=flow+t(flow)
      dis.msg=distances(msg)
         f.ee=c(d.flow[2,4]+d.flow[1,3]+d.flow[1,2]+d.flow[3,4],
d.flow[2,4]+d.flow[3,4]+d.flow[1,4],d.flow[1,3]+d.flow[3,4]+d.flow[2,3])
          if(any(f.ee>14400))
            {t=t+1
              next}
```

```
          cost[t,5]=sum(E(msg)$weight[f.ee>7200]*0.5)*1000/365
+sum(E(msg)$weight)*3000/365
+sum(dis.msg[lower.tri(dis.msg)==1]*d.flow[lower.tri(d.flow)==1])/1000
#####娴佽嘶璐圭敤
         print(t)
         t=t+1
       }

     }
   }
}
sum(cost[514,])
sort(rowSums(cost))+79500
order(rowSums(cost))
clas[c(187,190),]
cost[c(187,190),]
##########################################33
##########################3333333333333333

cost[is.na(cost)]<-Inf

total.cost=rowSums(cost)+7.95*10000

clas[c(513,514),]

min(total.cost,na.rm = T)
sort(total.cost)
min(rowSums(cost)+7.95*10000,na.rm = T)
which.min(rowSums(cost))
order(total.cost)
t=which.min(rowSums(cost))
#t=514
o=t
o
trans.ratio=un.ratio-
c(min(sum((need%*%y)[x==1][clas[t,]==1])/2,sum((data1[1,-
c(1:4)]%*%y)[x==1][clas[t,]==1]))/sum(data1[1,])
                    ,min(sum((need%*%y)[x==1][clas[t,]==2])/2,su
m((data1[2,-c(1:4)]%*%y)[x==1][clas[t,]==2]))/sum(data1[2,])
                    ,min(sum((need%*%y)[x==1][clas[t,]==3])/2,su
m((data1[3,-c(1:4)]%*%y)[x==1][clas[t,]==3]))/sum(data1[3,])
```

```
                                    ,min(sum((need%*%y)[x==1][clas[t,]==4])/2,su
m((data1[4,-c(1:4)]%*%y)[x==1][clas[t,]==4]))/sum(data1[4,]))
trans.ratio[trans.ratio<0]=0
trans.ratio
sum(trans.ratio)

oo=o-1
ic=c(oo%/%ro%/%ro%/%ro%%ro,oo%/%ro%/%ro%%ro,oo%/%ro%%r
o,oo%%ro)+1
ic

first=c(index[1,ic[1]],index[2,ic[2]],index[3,ic[3]],index[4,ic[4]])

##################################
i=ic[1];j=ic[2];k=ic[3];l=ic[4]

ii=c(i,j,k,l)
ftt=1
four=matrix(NA,100,3)
for(nn in 1:4)
{
   fir=as.matrix(data22[x==1,][clas[t,]==nn,4:5])
   di.2<-as.matrix(dist(fir),diag = F,upper=T)
   h=make_full_graph(dim(di.2)[1])
   E(h)$weight=di.2[lower.tri(di.2)==1]
   V(h)$name<-which(x>0)[clas[t,]==nn]
   ms<-mst(h)
   d=length(V(ms))
   edgelist=list()
   #


   for(ni in 1:d)
   {
      v=as.character(index[nn,ii[nn]])
      v.sh=shortest_paths(ms,v,weights = rep(1,(d-1)))$vpath[[ni]]$name
      sub.ms=induced_subgraph(ms,vids=as.character(v.sh))
      edgelist[[ni]]=t(get.edgelist(sub.ms))
   }
   edg=get.edgelist(ms)
   four.edge=matrix(0,d-1,3)
   ntt=1
```

```
f.e=0
for(ni in 1:nrow(edg))
{
   e.n=list()
   ntj=1
   for(nj in 1:d)
   {
      if(length(edgelist[[nj]])==0){next}
      if(any(apply(edg[ni,]==edgelist[[nj]],2,all)))
      {
         tab= table(as.numeric(edgelist[[nj]]))
         h=as.numeric(names(tab)[tab==1])
         e.n[[ntj]]=t(h[h!=index[nn,ii[nn]]])
         ntj=ntj+1
      }
   }
   if( length(e.n)>0)
      #[!duplicated(ena)]
      ena=as.numeric(as.data.frame(e.n))
   f.e=sum((need%*%y)[ena])
   if(f.e>14400)
   { break
   }
   if(f.e>7200)
   { four.edge[ntt,]=c(edg[ni,],ni)
   four[ftt,]=c(edg[ni,],ni)
   ftt=ftt+1
   ntt=ntt+1
   }
}
if(f.e>14400)
{ break}
#which(x>0)
el=as.matrix(as.data.frame(edgelist))
colnames(el)=1:ncol(el)
n.edge=NULL
for(np in 1:(d-1))
{
   n.edge[np]=sum(apply(get.edgelist(ms)[np,]==el,2,all))
}

dis.1to2=as.numeric(distances(ms,as.character(index[nn,ii[nn]]))/1000)
```

```
   dis.1toy=dist(data2[c(nn,index[nn,ii[nn]]+4),4:5])[1]/1000

c21=(sum(E(ms)$weight[four.edge[,3]]*0.5)+sum(E(ms)$weight*3))*1000/
365#####
   cost[t,nn]=c21+
      dis.1toy*4000000/365 ##    d
   + dis.1toy*(sum(data1[nn,])+sum(data1[,nn]))*0.5611
   +sum(dis.1to2*((need%*%y)[x==1][clas[t,]==nn]))
}
fo=matrix(four[!is.na(four)],,3)


######
ff=as.matrix(data22[c(index[1,i],index[2,j],index[3,k],index[4,l]),4:5])
di.1<-as.matrix(dist(ff),diag = F,upper=T)
g=make_full_graph(4)
E(g)$weight=di.1[lower.tri(di.1)==1]
V(g)$name<-as.character(c(index[1,i],index[2,j],index[3,k],index[4,l]))
msg<-mst(g)
cc=matrix(,4,110)
for (ni in 1:4) {
   cc[ni,]=rowSums(y[,x==1][,clas[t,]==ni])>0
}
flow=matrix(,4,4)
for(ni in 1:4)
{
   for(nj in 1:4)
   { flow[ni,nj]=(sum(data1[-c(1:4),-c(1:4)][cc[ni,],cc[nj,]])+
                             sum(data1[-c(1:4),][cc[ni],nj]))*0.1586+
      trans.ratio[ni]*sum(data1[,-c(1:4)][ni,cc[nj,]])
   }
}
diag(flow)=0
d.flow=flow+t(flow)
dis.msg=distances(msg)
f.ee=c(d.flow[2,4]+d.flow[1,3]+d.flow[1,2]+d.flow[3,4],
          d.flow[2,4]+d.flow[3,4]+d.flow[1,4],
          d.flow[1,3]+d.flow[3,4]+d.flow[2,3])
```

```
####################################################
########3333
####################################################
#################
###################
c1=list()
for(i in 1:4)
{c1[[i]]=which(x>0)[clas[o,]==i]
}
plot(data2[,4],data2[,5],type="n",asp=1,xlab = "x",ylab = "y")    ##empty plot
for(i in 1:4)
{
  fir=data22[x==1,][clas[o,]==i,4:5]
  di<-as.matrix(dist(as.matrix(fir),diag = F,upper=T))
  h=make_full_graph(dim(di)[1])
  graph_attr(h,'weight')=di[lower.tri(di)==1]
  sm<-mst(h,weights =graph_attr(h,'weight'))
  edg=get.edgelist(sm)
  which(first[1]==c1[[1]])
  lw=3
  if((sum(data1[i,])+sum(data1[,i]))*0.5611>14400 )
    lwd=6
  lines(data2[c(i,c1[[i]][which(first[i]==c1[[i]])]+4),4],
data2[c(i,c1[[i]][which(first[i]==c1[[i]])]+4),5],lwd=lwd)

  ## if()
  lwd=3
  for(j in 1:length(E(sm)))
  {
    lines(data22[c1[[i]][edg[j,]],4], data22[c1[[i]][edg[j,]],5],lwd=lwd)
  }
}
lwd=6
for(j in 1:nrow(fo))
{
  lines(data22[fo[j,1:2],4:5],lwd=lwd)
}


ff=as.matrix(data22[c(index[1,ic[1]],index[2,ic[2]],index[3,ic[3]],index[4,ic[
4]]]),4:5])
di.1<-as.matrix(dist(ff),diag = F,upper=T)
```

```
g=make_full_graph(4)
E(g)$weight=di.1[lower.tri(di.1)==1]
V(g)$name<-
as.character(c(index[1,ic[1]],index[2,ic[2]],index[3,ic[3]],index[4,ic[4]]))
msg<-mst(g)
m.dot=matrix(as.numeric(get.edgelist(msg)),,2)


for(nt in 1:nrow(m.dot) )
{    #if(f.ee[nt]>14400)
   lwd=3
   if(f.ee[nt]>7200 )    lwd=6
   lines(data22[c(m.dot[nt,]),4],data22[c(m.dot[nt,]),5],lwd=lwd)
}
points(data2[1:4,4],data2[1:4,5],pch=10,cex=3,col = 6)
for(i in 1:4)
{ #if(any(y[,y[,x==1]])

#points(data22[rowSums(y[,x==1][,clas[o,]==i])>0,4],data22[rowSums(y[,x
==1][,clas[o,]==i])>0,5],col=i+1)
   points(data22[first[i],4],data22[first[i],5],pch=17,cex=2.5,col=i+1)

points(data22[x==1,][clas[o,]==i,4],data22[x==1,][clas[o,]==i,5],col=i+1,pch
=16,cex=1.5)
}
```